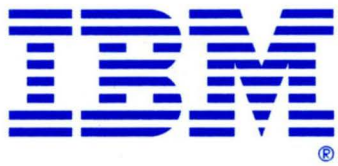
Volume 5, Number 3 May 1987

IBM Personal System/2 Models 50, 60, 80
Micro Channel™ Architecture,
Hardware Features and Design Considerations

IBM Personal System/2™ Seminar Proceedings

**The Publication for Independent Developers
of Products
for IBM Personal System/2**

**Published by International Business Machines Corporation
Entry Systems Division**



Changes are made periodically to the information herein; any such changes may be reported in subsequent Proceedings.

It is possible that this material may contain reference to, or information about IBM products (machines and programs), programming or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such products, programming or services in your country.

This publication could contain technical inaccuracies or typographical errors. Also, illustrations contained herein may show prototype equipment. Your system configuration may differ slightly. IBM believes the statements contained herein are accurate as of the date of publication of this document. However, IBM makes no warranty of any kind with respect to the accuracy or adequacy of the contents hereof.

This information is not intended to be a statement of direction or an assertion of future action. IBM expressly reserves the right to change or withdraw current products that may or may not have the same characteristics or codes listed in this publication. Should IBM modify its products in a way that may affect the information contained in this publication, IBM assumes no obligation whatever to inform any user of the modification(s).

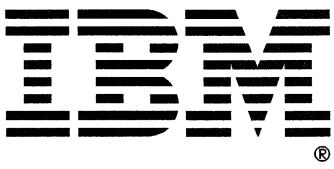
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever.

All specifications are subject to change without notice.

Printed in the
United States
of America

All Rights
Reserved

© Copyright International Business Machines Corporation 5/87



Contents

Foreword 1

Micro Channel Architecture 2
Introduction 2
Programmable Option Select 2
Introduction 2
POS Hardware Design 2
Sample Card Implementation 3
Channel Check Active Indicator 3
POS I/O Address Space Utilization 4
Sample Option Select Field Organization 4
Level-Sensitive Sharing Interrupts 4
Introduction 4
Card Design 5
Sequence of Operation - Overview 5
Multi-Device Arbitration Interface 5
Introduction 5
Direction 5
Design 6
Most Primitive Local Arbiter 6
Minimal Implementable Design 7
Local Arbiter Configuration 7
Local Arbiter Operation 7
Burst Mode Implementation 8
Preemption 9
Arbitration Bus Priority Assignments 9
Fairness 10
Linear Priority 11
Bus Time Out 12
Arbitration Latency and Minimum Bus Share Tutorial 12

Masters 14

Micro Channel 16
Description 16
Channel Definition 16
Signal Descriptions (16-Bit) 17
Signal Descriptions (32-Bit) 20
Signal Descriptions (Matched Memory) 21
Matched Memory Cycle 21
Signal Descriptions (Auxiliary Video Extension) 22
Micro Channel Connector (16-Bit) 23
Micro Channel Connector (Auxiliary Video Extension) 24
Micro Channel Connector (32-Bit Extension) 25
Micro Channel Connector (Matched Memory Extension) 25
Channel Signal Groups (16-Bit, 32-Bit, and Matched Memory) 26
Channel Signal Groups (Auxiliary Video Extension) 27
Micro Channel Critical Timing Parameters 27
Basic Transfer Cycle 27
I/O and Memory Cycle 28
DMA Timing 34
Arbitration Timing 40

Configuration Timing . . . 42
Auxiliary Video Connector Timing . . . 43
Adapter Design . . . 46
Physical Dimensions . . . 46
General Design Considerations . . . 46
Design Guidelines . . . 47

Feature Card Design Considerations . . . 49
Electro-Magnetic Compatibility (EMC) . . . 49
Card Form Factor . . . 49
Raw Card Thickness . . . 56
Assembled Card Thickness . . . 56
Primary Datum . . . 56
Examples . . . 56
Tab Pins - Necked Down . . . 56
Beveled Edge . . . 56
Internal Power Planes . . . 57
Example . . . 57
Internal Power Planes Into Tab Area . . . 58
Short Tabs on 5-Volt Pins . . . 58
Card Frame Ground/DC Ground Isolation . . . 59
Cards Without External Interface . . . 59
Cards With External Interface . . . 60
Analog Ground Isolation . . . 61
Card Layout With Metal Bracket . . . 62
Card Retainer . . . 67
Top Card Connectors . . . 68
External Connector Shielding . . . 69
Component Placement (Bus Interface) Micro Channel . . . 71
Thermal Considerations . . . 72
Power Availability Per I/O Slot . . . 72

Storage Subsystems . . . 73
Description . . . 73
DASD Subsystems . . . 73
Fixed Disks and Adapters . . . 73
Disk Cache . . . 74
3.5 Inch Diskette Drive and Media . . . 74
Media Compatibility and Transition Aids . . . 74
Description . . . 74
External Devices . . . 75
Description . . . 75
IBM 6157 Streaming Tape Drive . . . 75
IBM 3363 Optical Disk Drive . . . 75

Integrated Features . . . 76
DMA Function . . . 76
DMA Controller Transfer Modes . . . 76
General DMA Enhancements . . . 76
DMA Extended Mode . . . 77
Extended Mode Register . . . 77
Programming the DMA Controller Using Extended Mode . . . 77
Programming Example Using the Extended Mode . . . 78
Virtual DMA Channel operation . . . 78
PIO Function . . . 79
Timer Subsystem . . . 79
Programmable Option Select Logic . . . 79

Port B and NMI Logic. . . . 79
Alternate Processor Reset . . . 80
System Interrupts . . . 80
Nonmaskable Interrupt. . . . 80
Interrupt Assignments . . . 80
Numeric Coprocessor . . . 80
Serial Port . . . 80
Parallel Port . . . 81
Parallel Port Extended Mode . . . 81
Keyboard Controller . . . 81
Keyboard Password Security . . . 81
8042 Command and Status Bytes . . . 81
8042 COMMANDS (I/0 Address 064H) . . . 81
CMOS RAM . . . 82

IBM Personal System/2 Seminar Proceedings . . . 83

Foreword

IBM Personal System/2™ Seminars and Proceedings provide information about new product announcements and enhancements to existing products, and are intended to assist independent developers in their hardware and software development efforts.

Over the past several years, the success of the IBM Personal Computer family was due in part to the efforts of independent developers, whose hardware and software products have become widely used. For its part, IBM helped these vendors by holding relevant technical seminars and publishing the proceedings of those seminars. The result was a mutually beneficial partnership and transfer of technical knowledge.

This document is intended to assist the designer who is "Getting Started" with Micro Channel™ Architecture. The Seminar courses for which this text was developed are offered to respond to questions, extend the designers' perspective on systems design and assist acceptance of this new platform for growth by the industry. This document is also intended to assist those describing Personal System/2 to others, and to promote a general understanding of Micro Channel Architecture.

IBM's commitment to an open architecture is evident in the extensive and detailed Technical Reference manuals, seminars and technical publications. The optimization of Micro Channel for Surface Mount Technology (SMT) and Very Large Scale Integration (VLSI) technologies, coupled with a systems approach to hardware and software design, provides a stable foundation for development. The new architecture features of Program Option Select, Multi Device Arbitration, Level Sensitive Interrupts and Master Attachment Types, coupled with an objective for Personal Computer software compatibility create a *Platform For Growth*.

With the advent of the Personal System/2 family, IBM's seminar program will continue. Through these seminars and the corresponding Proceedings, IBM will address the independent developers' need for technical information about the latest IBM products. In these and future Proceedings, you will find technical information about subjects such as:

- IBM computer design and architecture
- IBM computer components and their interaction
- Memory capacities, speeds, transfer rates
- Input/output device capacities, speeds, access methods and rates
- Graphics and display technologies, programming considerations
- Printing technologies, programming considerations
- Operating system high level interfaces
- Development tools: capabilities, languages, program verification aids
- Compatibility considerations
- Communications: capabilities, offerings, statistics
- Enhancements to existing IBM hardware and software products
- Hints, tips and techniques to enhance your productivity

Through these seminars and Proceedings, IBM intends to maintain its partnership with independent developers and assist them in successfully producing hardware and software products for the IBM Personal System/2 Family.

Personal System/2 is a trademark of the IBM Corporation.

Micro Channel is a trademark of the IBM Corporation.

Micro Channel™ Architecture

Introduction

This section contains a tutorial overview of the new concepts of the Micro Channel Architecture:

- Programmable Option Select
- Level-Sensitive Interrupts
- Multi-Device Arbitration
- Master Attachment.

Because the physical size of the card and the card channel have changed, Micro Channel systems do not support existing Personal Computer adapters.

Refer to the *Personal System/2 Models 50, 60 and 80 Technical References* for detailed specifications, signal definitions and timing. In all areas of discrepancy with this document, the Technical Reference manual shall prevail.

Programmable Option Select

Introduction

The main objectives of Programmable Option Select (POS) are to:

1. Eliminate switches from the system board and feature cards.
2. Permit installation of multiple identical feature cards.
3. Positively identify any card by slot.
4. Resolve resource assignment conflicts.

It is desirable to eliminate the use of switches from the design in order to make the system more user friendly, permit direct customer setup, decrease documentation required for setup and improve serviceability.

Programmable Option Select (POS) replaces switches on the system board and the feature cards with software-writeable latches. Feature cards are identifiable by an ID number that the system reads and interprets. The system then writes the option selects to the option latches. System information is kept in a non-volatile RAM and is available to the setup routine at each power-up time.

The following benefits also result:

1. Card options may be altered to fit applications at the users' convenience.
2. Feature I/O and memory address space are not consumed excessively for initialization functions.

POS Hardware Design

Card Design

Since the physical size of the card and the card channel has changed, Micro Channel Architecture does do not support existing Personal Computer cards. Feature cards designed for use in Micro Channel systems (Models 50, 60 and 80) must comply with the following design rules:

1. Each card design decodes all 16 bits of the I/O address.
2. Each card design decodes all 24 bits of the memory address and MADE 24.
3. Each card design replaces the switches and jumpers with POS registers.
4. Each card implements POS registers that are selected by a unique slot select line and A0, A1, A2 address bits with latches.
5. Each card sends a card type ID to the data bus when interrogated. Only a "0" bit need be driven. This reduces the number of drivers required by many attachments. All data lines are resistor terminated to +5 for pullup.

   IBM assigns ID values for cards designed by or for IBM. ID values for non-IBM features are **not** assigned by IBM.

# Of Drivers ('0')	# Of Available ID Numbers
1	16
2	120
3	560
4	1820
5	4368
6	8008
7	11440
8	12870

Figure 1. Available ID Numbers

6. Each enabled card returns -CD SFDBK to the processor when an access is made to the card's address space. -CD SFDBK is not generated when the -CD SETUP line is in the setup state for addresses 0100-0107h.
7. Each card design is capable of de-gating all outputs to the system board (including CD SFDBK, DS16, interrupts, etc.) if the LSB of the option select word at 0102 is set inactive.
8. Following a reset, each card sets the bit 0 option select @ 0102 off.
9. If applicable, all cards may have the ability to reside at an alternate configuration (selectable using switches on PC/XT or Personal Computer AT).
10. Cards should fully decode A0, A1, and A2 with CD SETUP to select the register because CD SETUP is gated by a decode of the I/O address range 0100 - 0107 by the system.

    Alternate configurations are highly desirable to maximize the number of possible system configurations (combinations of features). It is recommended that one or two alternate I/O selects be provided. Note that each card selected by arbitration level, such as a DMA slave, must be programmable to arbitration levels 0-15 through POS (the configuration utilities will not select arbitration level 15 due to conflict with the processor). Similarly, any card which implements ROM (or RAM) that resides in the BIOS ROM area must support relocation of the ROM space to any of 16 start addresses by POS.

Sample Card Implementation

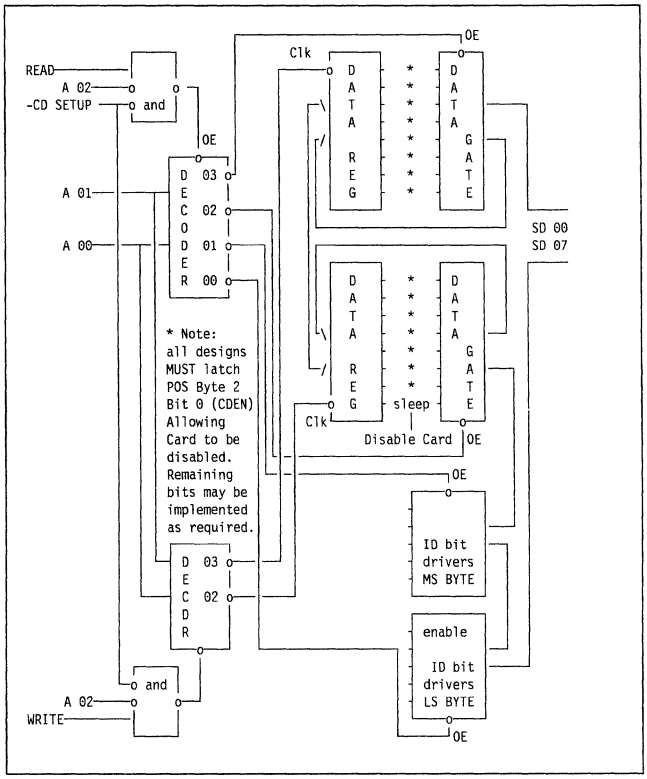


Figure 2. Sample Card Implementation of POS

1. READ and WRITE are decoded from S0, S1 and timed by CMD
2. -CD SFDBK is not activated during setup cycles
3. The card needs only to decode the three least significant bits of the address.

The system will gate CD SETUP with an I/O decode of 0100h to 0107h.

Channel Check Active Indicator

Micro Channel memory and I/O functions that report Channel Check (CH CK) must set an active indicator to identify the source of the error. This indicator is at POS REG 5 bit 7 of each card's option select address space. This bit is interrogated by the NMI handler responding to CH CK for each card position until all reporting cards have been identified. Figure 3 on page 4 shows a sample implementation:

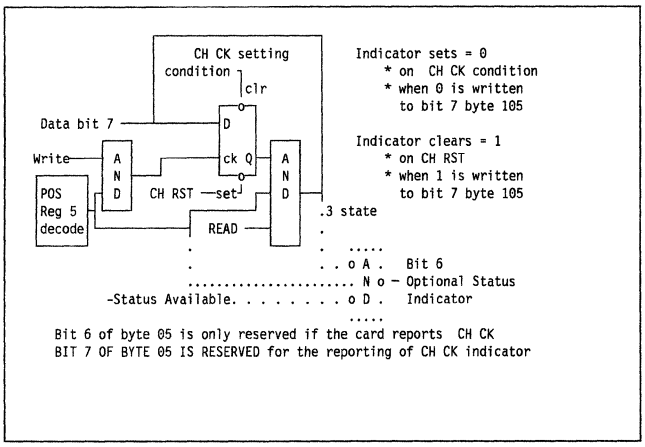

CH CK setting condition
clr
Data bit 7
D
Write
A N D
ck Q
CH RST
set
POS Reg 5 decode
READ
.3 state
Indicator sets = 0
* on CH CK condition
* when 0 is written to bit 7 byte 105
Indicator clears = 1
* on CH RST
* when 1 is written to bit 7 byte 105
Bit 6
Optional Status Indicator
-Status Available.
Bit 6 of byte 05 is only reserved if the card reports CH CK
BIT 7 OF BYTE 05 IS RESERVED for the reporting of CH CK indicator

Figure 3. CH CK Active/Status Available Indicator

The channel check signal may be reset at any point during the CH CK service routine, but must be reset prior to exiting the routine.

POS I/O Address Space Utilization

The software can access the following eight reserved byte positions in the feature's address space. Each card decodes the three least significant bits of the address bus and -CD SETUP.

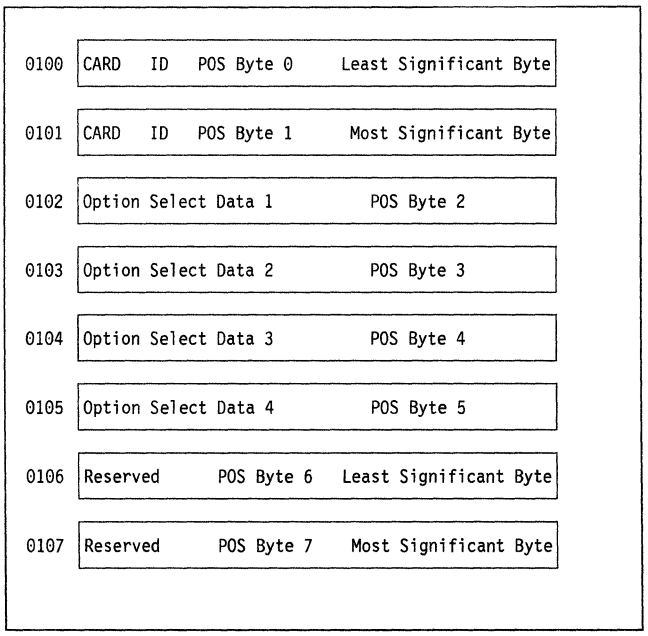

0100 CARD ID POS Byte 0 Least Significant Byte
0101 CARD ID POS Byte 1 Most Significant Byte
0102 Option Select Data 1 POS Byte 2
0103 Option Select Data 2 POS Byte 3
0104 Option Select Data 3 POS Byte 4
0105 Option Select Data 4 POS Byte 5
0106 Reserved POS Byte 6 Least Significant Byte
0107 Reserved POS Byte 7 Most Significant Byte

Figure 4. POS Address Space Utilization

Sample Option Select Field Organization

Only the least significant bit of the option select byte at address 102 and the two most significant bits at address 105 are architected. All other fields within the address range 102 to 105 are free-form. The following is a sample orientation for POS bits.

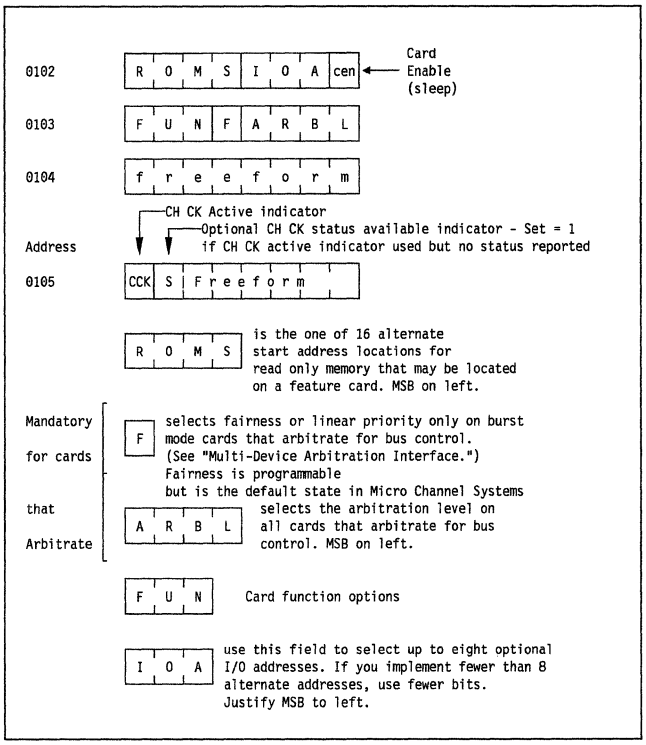

0102 ROMS IOA cen
Card Enable (sleep)
0103 FUNF ARBL
0104 freeform
CH CK Active indicator
Optional CH CK status available indicator - Set = 1
if CH CK active indicator used but no status reported
Address
0105 CCK S Freeform
ROMS is the one of 16 alternate start address locations for read only memory that may be located on a feature card. MSB on left.
Mandatory for cards that Arbitrate
F selects fairness or linear priority only on burst mode cards that arbitrate for bus control. (See "Multi-Device Arbitration Interface.") Fairness is programmable but is the default state in Micro Channel Systems
ARBL selects the arbitration level on all cards that arbitrate for bus control. MSB on left.
FUN Card function options
IOA use this field to select up to eight optional I/O addresses. If you implement fewer than 8 alternate addresses, use fewer bits. Justify MSB to left.

Figure 5. Default Option Select Field Organization.

Level-Sensitive Sharing Interrupts

Introduction

The main objectives of Level-Sensitive Sharing Interrupts are to:

1. Simplify sharing the logic on feature card and system board designs
2. Reduce transient sensitivity of the interrupt controller
3. Provide compatibility with existing software.

The side benefits that result are:

1. A mixture of sharing and non-sharing hardware on the same level
2. Interrupt sharing between the system unit and expansion features

Card Design

All Micro Channel system board features and channel attach features employ the same level-sensitive, negative-active interface mechanism. Each card implements an open collector driver (or tri-state driver gated negative active) to drive the interrupt request line for the level assigned to the card function.

Each card also implements an interrupt pending indicator as a bit position in the I/O address space. All cards must provide an interrupt pending latch (driving the interrupt pending indicator) that is reset by normal servicing of the device.

All devices are now defined as sharing on all levels, and all devices are capable of responding to software that does not reset the sharing logic, so no problem exists with this scheme when devices are mixed randomly on a level. Service latency will increase as the number of shared devices per level is increased.

A card must hold the level-sensitive interrupt active until it is reset as a direct result of servicing the interrupt. Service routines shall not attempt to end the interrupt sequence (EOI) until it has reset the interrupt line of the device being serviced.

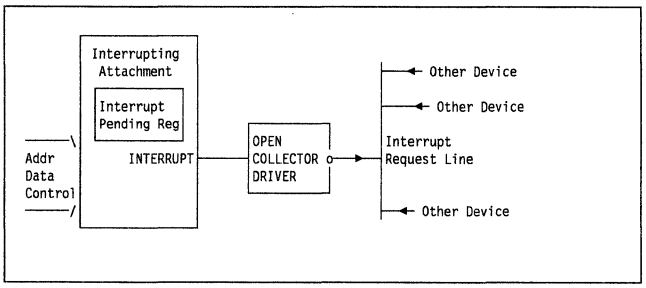


Figure 6. Typical Attachment Implementation

Sequence of Operation - Overview

Level-sensitive interrupts are interlocked between the hardware and software that support the interrupt service. For this reason, phantom or lost interrupts are typically less frequent and more easily isolated.

Note: The interrupt controller must not be reset (EOI) until **after** the interrupt line has been reset on the channel.

Figure 7 shows a simplified interlocked sequence:

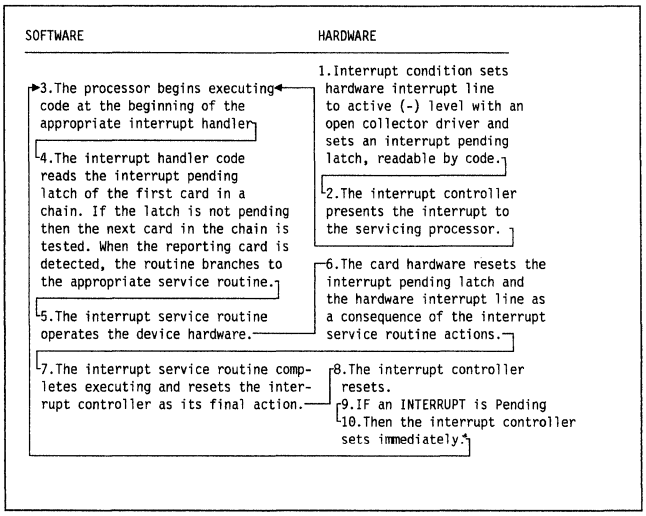

SOFTWARE
HARDWARE
1.Interrupt condition sets hardware interrupt line to active (-) level with an open collector driver and sets an interrupt pending latch, readable by code.
2.The interrupt controller presents the interrupt to the servicing processor.
3.The processor begins executing code at the beginning of the appropriate interrupt handler.
4.The interrupt handler code reads the interrupt pending latch of the first card in a chain. If the latch is not pending then the next card in the chain is tested. When the reporting card is detected, the routine branches to the appropriate service routine.
5.The interrupt service routine operates the device hardware.
6.The card hardware resets the interrupt pending latch and the hardware interrupt line as a consequence of the interrupt service routine actions.
7.The interrupt service routine completes executing and resets the interrupt controller as its final action.
8.The interrupt controller resets.
9.IF an INTERRUPT is Pending
10.Then the interrupt controller sets immediately.

Figure 7. Level-Sensitive Interrupt Sequence with Sharing

Multi-Device Arbitration Interface

Introduction

The main objectives of the Multi-Device Arbitration Interface are to:

1. Allow additional DMA device(s)
2. Allow burst data transfers
3. Allow prioritization of control between DMA devices
4. Support multiple masters.

Direction

The IBM Micro Channel provides a high-performance Multi-Device Interface with the ability to support up to 15 arbitrating devices. The Multi-Device Interface supports DMA features and masters, prioritizes their access to control of the channel, and provides burst capability with both preemption and fairness features. The arbitration delays are significantly reduced over previous designs.

The side benefits that result are:

1. The number of signal lines required to support DMA transfers is now reduced on the channel interface and to LSI packages.
2. Decentralized bus control arbitration allows decentralized implementation of interface control logic.
3. Misassignments in arbitration level result in a loss of performance rather than a less graceful overrun condition.

Design

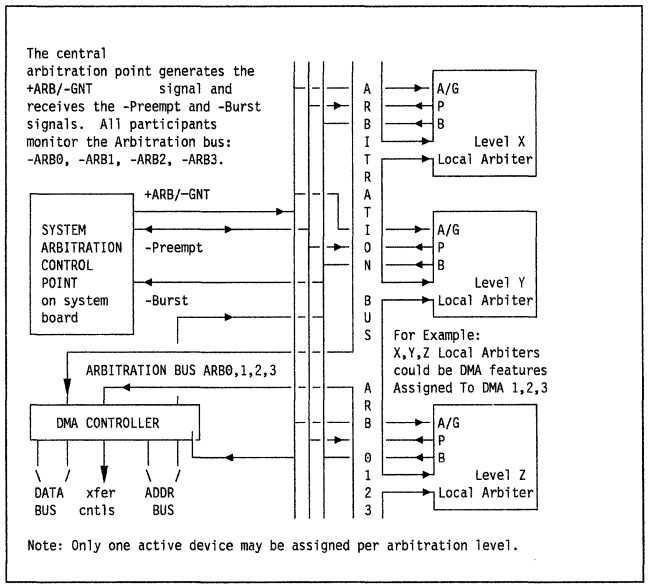
Interface Description

Seven lines comprise the Multi-Device Interface:

+ARB/-GNT
-PREEMPT
-BURST
ARB0 ┐
ARB1 ├ Arbitration
ARB2 │ Bus
ARB3 ┘

Lines 4, 5, 6 and 7 represent the arbitration bus.

The seven lines coordinate arbitration logic that is distributed among all devices from a single arbitration point on the system board.


The central arbitration point generates the +ARB/-GNT signal and receives the -Preempt and -Burst signals. All participants monitor the Arbitration bus: -ARB0, -ARB1, -ARB2, -ARB3.
+ARB/-GNT
SYSTEM ARBITRATION CONTROL POINT on system board
-Preempt
-Burst
ARBITRATION BUS ARB0,1,2,3
DMA CONTROLLER
DATA BUS
xfer cntls
ADDR BUS
ARBITRATION BUS
A/G
P
B
Level X Local Arbiter
Level Y Local Arbiter
Level Z Local Arbiter
For Example: X,Y,Z Local Arbiters could be DMA features Assigned To DMA 1,2,3

Note: Only one active device may be assigned per arbitration level.

Figure 8. Multi-Device Arbitration Interface

Most Primitive Local Arbiter

The simplest implementation of the interface is a device that waits patiently to make a single transfer. The sequence of events would be as follows. (Time is not a linear scale, and the operation is spaced for clarity.)

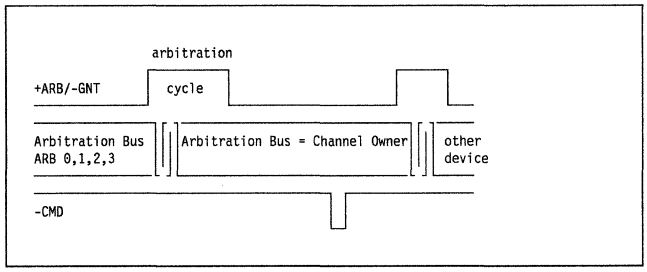


Figure 9. Arbitration Timing

1. When the +ARB/-GNT line goes to the arbitrate state, each device that wants the bus places its assigned arbitration level on the arbitration bus. For example, ARB0=1, ARB1=0, ARB2=1, and ARB3=1 when the device on arb level 2 places its bid for control on the arbitration bus.
2. If a value lower than the device's assigned value is placed on the bus by another device, then the higher-valued (lower priority) device must remove the lower-ordered bits of its bid. The circuitry to do this logical operation is surprisingly simple.
3. The System Arbitration Control Point that generates the +ARB/-GNT signal, times out a period sufficient to stabilize the arbitration bus and returns the +ARB/-GNT signal to the grant state.
4. A device recognizes when it has received the grant by combinatorially decoding the arbitration bus and the +ARB/-GNT signal and enabling the device interface as +ARB/-GNT returns to the grant state. The arbitration bus is stable throughout the grant state. If +ARB/-GNT goes to the arbitrate state, the device *must degate all drivers to the channel and cease all channel operations immediately*!
5. After receiving the grant, a device may "own" the channel until the trailing edge of the next memory or I/O read or write.
6. TC (Terminal Count) points to the device presently owning the channel.
7. Control of the channel then returns to the System Arbitration Control Point and another arbitration cycle results.

Minimal Implementable Design

The previous implementation relies on arbitration cycles that are periodically generated by the System Arbitration Control Point. If those cycles are not inserted on a demand basis, but periodically as the primitive design requires, a general performance impact is implied as arbitration cycles would be inserted unnecessarily. That mechanism is therefore parasitic and not supportable without a line to request an arbitration cycle. Such a function is called "Preempt" and produces the "Implementable Design" sequence shown in Figure 10.

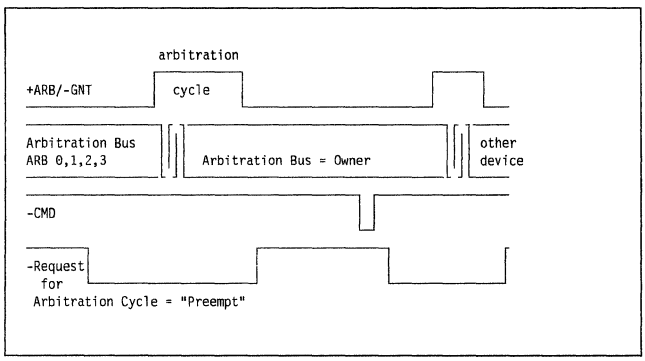


Figure 10. Minimal Implementation of Multi-Device Interface.

Notes:

1. Arbitration cycles are inserted by the System Arbitration Control Point **only** when a request is received from an arbitrating device or when the channel is returned to the default owner (the system board processor).
2. To insure integrity, the requesting device holds the request for arbitration cycle line (-PREEMPT) active until after it has received the grant. The request is removed after the falling edge of +ARB/-GNT.

It can be seen, therefore, that a minimal implementation of a simple channel device would require at least support of the lines indicated in Figure 10. The request for arbitration function is also needed to support more advanced functions on the channel; it is combined with those functions to yield the -PREEMPT signal. The -PREEMPT signal is a request for an arbitration cycle, and more. The -PREEMPT signal is further described under "Preemption" on page 9.

Local Arbiter Configuration

In order to keep arbitration cycles as short as possible, local arbiters must use feed forward (parallel arbiter designs). This means that during arbitration, when your arbiter loses a given bit of the arb bus, all lower bits must be turned off immediately. A simplified arbiter configuration is shown in Figure 11.

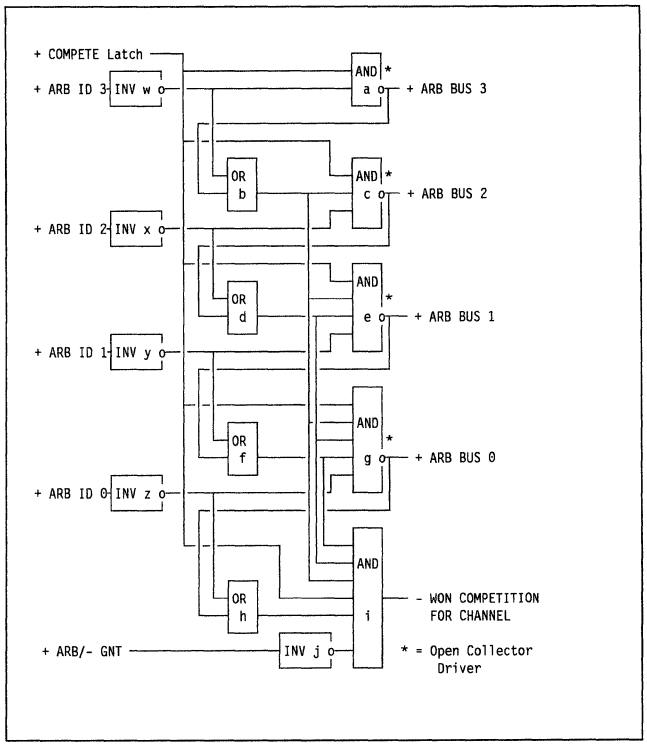


Figure 11. Simplified Local Arbiter

Local Arbiter Operation

The Local Arbiter circuit diagram does not show the control logic for -PREEMPT, -BURST and programmable fairness. Only the interface to the arbitration bus is shown. The Arb ID set by POS [ARBL] is 6 or A3,2,1,0 = 0110. This value is inverted to 1001 by inverters w,x,y,z. The sequence of operation is as follows against an arbiter elsewhere with an assigned arbitration level of 7 or A3,2,1,0 = 0111.

1. The Arb ID set by POS (ARBL) is stored negative as 0111.
2. The device sets -PREEMPT and waits for an arbitration cycle.
3. With pending preemption and the positive transition of ARB/-GNT, the Compete Latch is set (Request).
4. The +ARB/-GNT line goes to the Arbitrate state.

5. The opposing arbiter sets A3 = 0. A2 = 1, A1 = 1, and A0 = 1.
6. The NAND gate "a" drives +ARB3 low, because COMPETE = 1 and -ARB ID bit 3 = 1.
7. The OR gate "b" does not disable lower-ordered stages, because one of its inputs is high (the -ARB ID bit 2 is high). If neither of the inputs to this gate were 1 then all lower ordered NAND drivers would be disabled.
8. NAND gate "c" sets its output high, because the -Arb ID bit 2 is low.
9. The OR gate "d" does not disable lower-ordered stages, because one of its inputs is high (the +ARB2 bit is high on the arbitration bus). If this bit on the arbitration bus were driven low by an opposing arbiter then all lower-ordered stages of this arbiter would be disabled, and this arbiter would withdraw from competition.
10. NAND gate "e" sets its output open (high) because the -Arb ID bit 1 is low.
11. The OR gate "f" does not disable lower ordered stages because one of its inputs is high (the +ARB1 bit is high on the arbitration bus).
12. NAND gate "g" drives +ARB3 low because COMPETE = 1 and -ARB ID bit 3 = 1 and it is not disabled by a higher-ordered compare.
13. The OR gate "h" does not disable lower-ordered stages, because one of its inputs is high (the -ARB ID bit is high).
14. The opposing arbiter loses the competition at this point because its +ARB ID bit 0 is high and +ARB0 is low. Both inputs to its OR gate "h" are low, and the AND gate "i" in the opposing arbiter is disabled.
15. After a prescribed settling time, +ARB/-GNT goes low to the grant state.
16. Because no stage in the arbiter has disabled an input to AND gate "i", and because +ARB/-GNT is low, and because COMPETE is high, the AND gate "i" is high. The arbiter signals BUS WON (Acknowledge).
17. As long as +ARB/-GNT is low the arbitration bus is stable and the BUS WON signal is stable. If +ARB/GNT goes high, BUS WON goes inactive within 2 gate delays; the COMPETE latch will not be set unless preempt is driven by the arbiter.

Burst Mode Implementation

Some features typically transfer data in bursts that are often separated by long quiescent periods. A disk file is an example of such a feature; when transferring a track, the entire block of data is spun off at one time. Sectors may be arranged one after the other with little or no quiescent time between them, but with long track-step latencies exist between whole track block transfers. Typically such devices are buffered to accept one or more blocks of data in a giant gulp that is later transferred byte-by-byte across the channel. This procedure has the built-in delay imposed by the buffer on loading and unloading, as well as a delay encountered on the channel as the attachment contends for service. These delays affect performance, and the buffer circuitry affects product cost. If the entire block can be transferred to storage in a few relatively large chunks or (typically) as one piece, the cost and delay of buffering can be minimized. This is one objective of burst mode. Burst mode also reduces arbitration per transfer by moving whole blocks per arbitration cycle; it also reduces attachment cost and increases performance.

The attachment activates the burst line upon acquiring the grant and holds it active until after the leading edge of the last card pulse. The burst line is released after the leading edge of the last command pulse in the burst sequence. The Central Arbitration Point will not produce arbitration cycles when another device requests the channel. This is in effect saying, "I will take every cycle, if you'll let me."

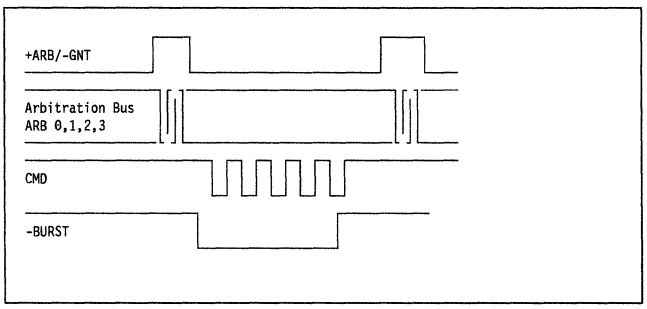


Figure 12. Burst Mode

Note: It is permissible for an arbiter to set burst for a single transfer, because the attachment may be unaware of the last transfer until it is upon it.

Preemption

Burst mode is fine for the feature that is bursting. The situation is analogous to two scuba divers, who are buddy breathing: It's OK if your buddy takes a breath or two, but if he takes 200 in a row, then you're in trouble. The way that a device can let the Central Arbitration Point know that it is running out of air is to activate the -PREEMPT line. This line is also the request for an arbitration cycle alluded to in "Minimal Implementable Design" on page 7. It can be activated by any device that desires service, regardless of that device's assigned priority level. Momentarily assume that the requesting device is of a higher priority. When the arbitration cycle occurs, the higher priority device will win the grant. This is shown is Figure 13:

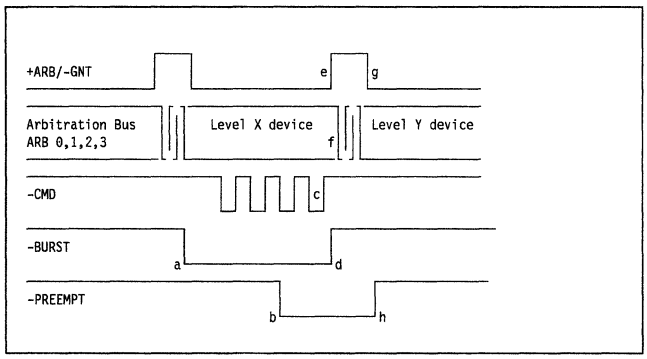


Figure 13. Burst Mode with Preemption

The sequence of events in the example is as follows:

1. A burst mode device gains control of the channel (a).
2. A device nearing an overrun condition requests preemption (b).
3. The current owner recognizes -PREEMPT (c) and prepares to surrender the channel by completing any partial transfers. Partial transfers are CMD cycles that occur after the external preemption that allow the device to terminate in a desirable state.
4. The device that is losing the grant removes the burst line (d) and does not participate in the bus arbitration cycle (d).
5. The Central Arbitration Point removes the grant (e).
6. Arbitration for bus control begins (f).
7. After +ARB/-GNT fails, control of the bus is granted to a new device (g).
8. The preempting device removes -PREEMPT in response to the grant (h).

Arbitration Bus Priority Assignments

All arbitrating devices on the channel are prioritized as shown in Figure 14:

Arbitration Level	Primary Assignment	DMA Assignment
-2	Memory Refresh	----------
-1	Error Recovery	----------
0	DMA Port 0	Programmable
1	DMA Port 1	Fixed
2	DMA Port 2	Fixed
3	DMA Port 3	Fixed
4	DMA Port 4	Programmable
5	DMA Port 5	Fixed
6	DMA Port 6	Fixed
7	DMA Port 7	Fixed
8	SPARE	
9	SPARE	NOTE: DMA Ports
A	SPARE	0 and 4 can be
B	SPARE	programmed to
C	SPARE	respond to any
D	SPARE	arbitration level
E	SPARE	
F	System Board Processor	
ISQ	Inactive State Queue (HOGPEN) (Explained in following sections)	

Figure 14. Arbitration Priority

The DMA controller has two channels that respond to a programmable arbitration level. Following a reset, one port is assigned to port 0 and the other port is assigned to port 4. Using this feature, up to 15 DMA devices can be supported from the system board DMA controller by reassigning the programmable channels as required. The additional DMA devices are supported by "virtual" channels.

Typically a driver or BIOS for the DMA controller is installed to funnel requests for a virtual channel to the programmable DMA channels and maintain the busy/in use status and the assigned DMA I/O control port addresses for the requesting slave device's BIOS.

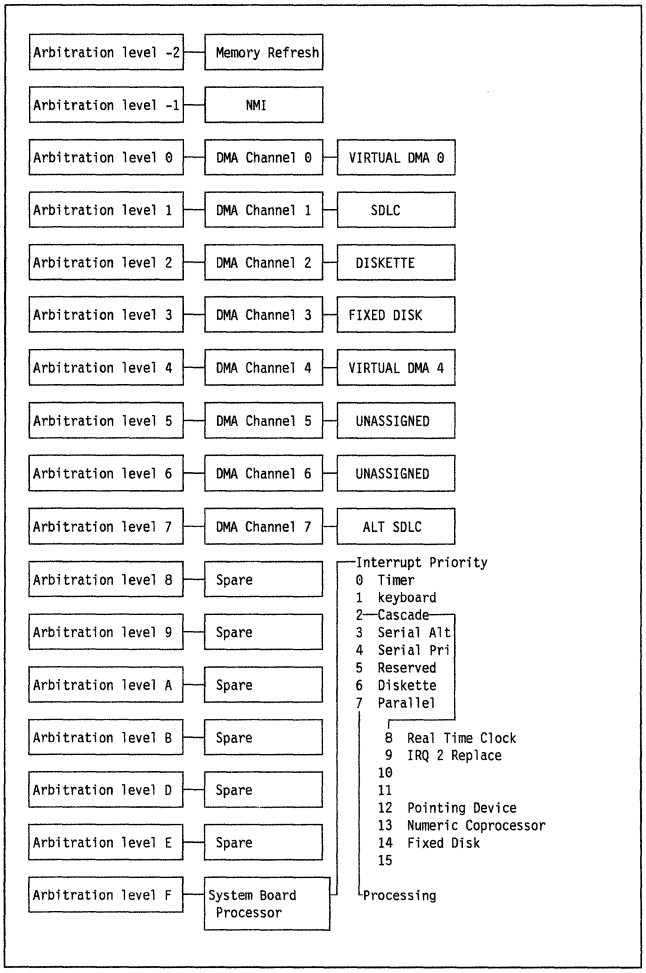


Figure 15. System Priority Map

Fairness

Readers with a sense of democracy have noticed that a high-bandwidth, high-priority device that has received the grant, would give the channel up for one cycle, but would gain it back on the next. This is fine if there are no other devices contending for ownership of the channel, but if a second or third device with low priority wants control, it would still be locked out. There is fairness on the Micro Channel.

Previous Personal Computer designs that did not provide burst operations depended on the single transfer mode to do all DMA operations. Equity or sharing of the channel was guaranteed due to the opportunity for channel requests that existed between the DMA transfers of even the faster devices. Burst operations give up this "throttling" characteristic and run the risk of "hogging" or inequitable sharing of the channel access.

To include a form of equity in burst operations, the Multi Device Interface implements a fairness algorithm that guarantees each device a share of the channel in a sequence determined by each device's priority. There is a maximum latency for access to the channel. For a given priority and configuration, there is a minimum guaranteed performance level. Quite simply, if a device loses the grant, it is temporarily declared a "hog," and must wait until after the (common) -PREEMPT line goes (+) inactive and either status bit goes (+) inactive before competing for the channel again.

In practice, as devices queue up for channel control, the "dot-ored" -PREEMPT will remain active until all devices have been served in priority sequence. Hogs will limit their ownership of the channel to separated bursts of transfers, but will continue to operate.

Fairness Algorithm for Bursting Devices

All devices on the Micro Channel that implement burst move, must also implement the fairness algorithm: if you lose the grant, you must not compete for channel control until after the -PREEMPT line goes (+) inactive and either status bit makes an active to inactive (- to +) transition.

Example:

Burst devices on levels 2,3,6,9 have preempted each other in an indeterminate sequence. A fifth device, that presently owns the bus, preempted the last of the devices that entered the "Hog Pen". So the fifth device is the last driver pulling down Preempt and Preempt goes (+)inactive.

Devices in the hog pen remain inactive until after -PREEMPT goes + and either STATUS goes +.

When the fifth device removes Preempt from the channel, all the devices exit the hog pen and join in arbitration together.

The algorithm thereby waits until all contestants have had a shot at the channel before repeating the cycle. The algorithm apportions the channel according to the number of partial transfers the attachment inserts before completing activity on the channel and the arb level.

Preemption

Burst mode is fine for the feature that is bursting. The situation is analogous to two scuba divers, who are buddy breathing: It's OK if your buddy takes a breath or two, but if he takes 200 in a row, then you're in trouble. The way that a device can let the Central Arbitration Point know that it is running out of air is to activate the -PREEMPT line. This line is also the request for an arbitration cycle alluded to in "Minimal Implementable Design" on page 7. It can be activated by any device that desires service, regardless of that device's assigned priority level. Momentarily assume that the requesting device is of a higher priority. When the arbitration cycle occurs, the higher priority device will win the grant. This is shown is Figure 13:

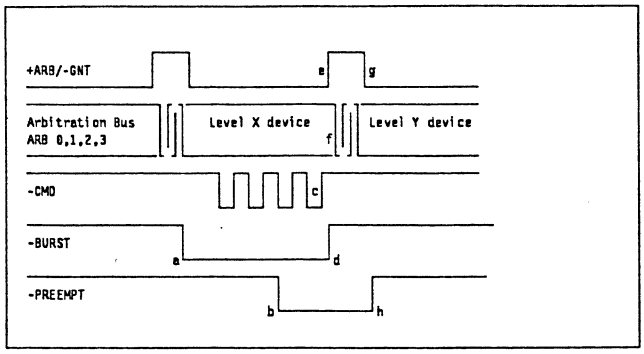


Figure 13. Burst Mode with Preemption

The sequence of events in the example is as follows:

1. A burst mode device gains control of the channel (a).
2. A device nearing an overrun condition requests preemption (b).
3. The current owner recognizes -PREEMPT (c) and prepares to surrender the channel by completing any partial transfers. Partial transfers are CMD cycles that occur after the external preemption that allow the device to terminate in a desirable state.
4. The device that is losing the grant removes the burst line (d) and does not participate in the bus arbitration cycle (d).
5. The Central Arbitration Point removes the grant (e).
6. Arbitration for bus control begins (f).
7. After +ARB/-GNT fails, control of the bus is granted to a new device (g).
8. The preempting device removes -PREEMPT in response to the grant (h).

Arbitration Bus Priority Assignments

All arbitrating devices on the channel are prioritized as shown in Figure 14:

Arbitration Level	Primary Assignment	DMA Assignment
-2	Memory Refresh	----------
-1	Error Recovery	----------
0	DMA Port 0	Programmable
1	DMA Port 1	Fixed
2	DMA Port 2	Fixed
3	DMA Port 3	Fixed
4	DMA Port 4	Programmable
5	DMA Port 5	Fixed
6	DMA Port 6	Fixed
7	DMA Port 7	Fixed
8	Reserved	
9	Reserved	NOTE: DMA Ports
A	Reserved	0 and 4 can be
B	Reserved	programmed to
C	Reserved	respond to any
D	Reserved	arbitration level
E	Reserved	
F	System Board Processor	
ISQ	Inactive State Queue (HOGPEN) (Explained in following sections)	

Figure 14. Arbitration Priority

The DMA controller has two channels that respond to a programmable arbitration level. Following a reset, one port is assigned to port 0 and the other port is assigned to port 4. Using this feature, up to 15 DMA devices can be supported from the system board DMA controller by reassigning the programmable channels as required. The additional DMA devices are supported by "virtual" channels.

Typically a driver or BIOS for the DMA controller is installed to funnel requests for a virtual channel to the programmable DMA channels and maintain the busy/in use status and the assigned DMA I/O control port addresses for the requesting slave device's BIOS.

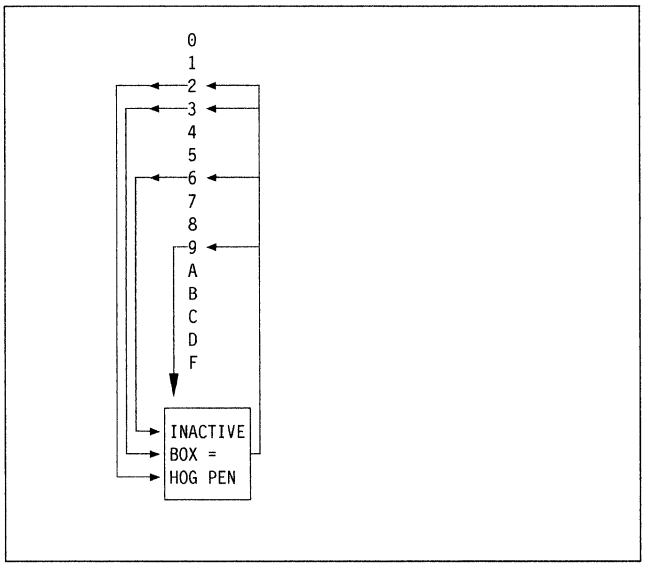


Figure 16. Fairness Algorithm

Fairness Timing

The devices that implement burst mode have a disable latch that holds them inactive if the negative level of -PREEMPT is encountered while their burst line is active. The latch is reset by inactive -PREEMPT and an active to inactive transition on either status bit.

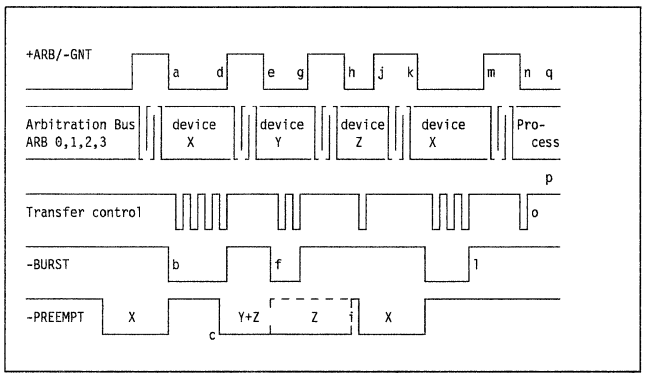


Figure 17. Complex Arbitration with Queued Preemption

The sequence of events is:

1. Device X gains control of the channel (a).
2. Device X begins burst operation (b).
3. Devices Y and Z preempt the burst operation of device X (c).
4. Device X releases burst and arbitration begins; device X enters the Inactive State Queue (hog pen) (d).
5. Device Y captures the channel because Y is at a higher priority than Z (e).
6. Device Y begins burst operation (f)
7. Device Y gives up the channel after completing partial transfers (g), and enters the Inactive State Queue (hog pen).
8. Device Z captures the channel (h)
9. Device Z releases -PREEMPT line (i).
10. Devices X and Y exit Inactive State Queue (hog pen)
11. Device Z releases the channel after an incremental cycle steal operation (j).
12. Devices X and Y compete in arbitration.
13. Device X regains the channel to continue its burst operation (k).
14. Device X removes the burst tag after the leading edge of the last CMD(1).
15. Without a pending request, the channel defaults to the processor, and an arbitration cycle is inserted by the central arbitration point(m).
16. The lowest priority device, the processor, gets the channel (n).
17. The processor fetches an instruction (o).
18. The processor continues the instruction (p).
19. The channel is retained after the instruction has completed, if -PREEMPT is inactive (q).

Linear Priority

Another mode can be defined easily by altering two of the arbitration rules. This mode defeats fairness and allows some or all devices to legitimately hog the channel. This is a programmable feature; the programmability allows fine tuning of a system for a given application or the use of feature cards in other systems where fairness is not desirable. The default for Micro Channel is Fairness Mode.

The two arbitration rules that are changed are:

1. Item 4 on page 9 is changed to:

   "The device that is losing the grant now MAY participate in the next arbitration cycle."

2. Item 4 is changed to:

   "Device X releases burst and arbitration begins; device X does not enter the Inactive State Queue (hog pen) (d)."

   This means that if a preempt is detected on the channel by the active device, that device must complete all partial transfers and exit the

channel. The device may compete in the next arbitration cycle and may regain control of the channel if it has the highest priority arbitration level.

The reset condition for all arbiters is fairness. Linear priority shall be selected only as a Programmable Option Select parameter. Arbiters on the multi-device arbitration interface shall support programmability of fairness and of all 16 arbitration levels.

Bus Time Out

If a burst mode device were to gain control of the channel and refuse to release control, refresh operations might be impeded, causing soft memory errors.

To protect the machine against such devices, a channel time-out feature has been added to the central arbitration point. *All burst devices are allowed a maximum of 7.5 microseconds of activity following activation of PREEMPT, whether or not CMD transfers occur.* This means that a device is guaranteed a minimum of 7.5 microseconds of activity on the channel (if burst is held active, of course). Furthermore, the period of channel ownership will be longer, because the refresh circuity will set Preempt only once every 15.6 microseconds. By monitoring the -PREEMPT line, a burst mode arbiter will get an average of about 12 microseconds of activity. In other words, refresh may set -PREEMPT as late as 15.6 microseconds after the channel is granted to a burst mode arbiter, or as early as zero microseconds after the channel is granted. *In no case may an arbiter hold onto the channel longer than 7.5 microseconds after active PREEMPT.*

The central arbitration control point will not grant the channel to an arbiter when a refresh is due to occur.

If a device holds burst active longer than 7.5 microseconds after active Preempt, the central arbitration point will place the +ARB/-GNT line in the Arbitrate state, thereby removing the grant. This should degate the combinatorial decode of the grant to the local arbiter regardless of the consequences to the arbitrating device. Remember that during arbitration, the +ARB/-GNT line will be high and no device except the processor may be active.

In a sense, the device does not own the channel, it simply leases it for a period of 7.5 microseconds. The landlord will give notice to pay the rent and if the tenant continues to reside on the channel, the tenant will be evicted.

Arbitration Latency and Minimum Bus Share Tutorial

This section is provided to demonstrate the effect of the fairness algorithm. Timings in the example are not necessarily indicative of any system.

Fairness guarantees all channel participants a share of the bus according to the following rules defining latency and a share of the channel to a preempting device.

- Maximum worst case time to free the channel from the present owner (processor for certain situations) is 23μs. release time. Typical processor release time is less than 12 μs.
- Arbitration request latency will be approximately 16 μs worst case per arbiter in line. (sum of 7.5 μs + refresh + arbitration + clock synchronization + guardband)
- Total latency is release time plus arbitration request latency.
- Non-burst arbiters will get (max) single cycles at the rate of request.
- Burst arbiters will transfer a (max) number of cycles that is the product of the number of channel grants and the number of partial transfers allowed per grant.
- The ratio of channel grants per arbiter, with two continuously bursting fairness arbiters competing, will be 50:50.
- If more than two bursting fairness arbiters compete, then, theoretically, for n arbiters:

The nth and (n-1)th arbiters will each get $\frac{1}{2-1}$ of the bus min.

The first through (n-2)nd arbiters will get $\frac{1}{n-1}$ of the bus min.

For Example:
Assuming time per transfer is equal for all arbiters

Arbiter #	Assigned Arb Level	Allowed Number of Partial Transfers	Percent Grants	Bus Share min Percent	Latency Total
1	0	1 / Grant	20	20/400= 5%	23μs
2	2	5 / Grant	20	100/400=25%	55μs
3	3	3 / Grant	20	60/400=15%	71μs
4	5	1 / Grant	20	20/400= 5%	103μs
5	A	19 / Grant	10	190/400=47.5%	183μs
6	C	1 / Grant	10	10/400= 2.5%	215μs

Note that with fairness, all arbiters get a guaranteed minimum share. Linear priority can give all of the channel to arbiter 1 in a contest.

* This is the first time that latency, successive latencies will be rotational calculated as the sum of all other device latencies.

Figure 18. Arbitration Latency and Bus Share

Masters

The Bus Master attachment type is supported by the Channel Architecture. This attachment type offers advantages and disadvantages compared to a DMA slave:

1. Cost
2. Complexity and timing constraints
3. D. C. power and heat
4. Random or sequential operations on storage and I/O
5. Consumes 1/2 to 1/4 of the channel bandwidth of DMA
6. High-Level functionality
7. Compatibility with a DMA slave
8. Extension of system performance.

A master must drive the address, data, transfer control and arbitration channels, and therefore can consume larger amounts of power from driver modules alone. Typically the design point is chosen to allow a processing unit or "captured" DMA controller to access the channel to gain performance or reduce bandwidth. These control elements also consume power.

A master must monitor the CHRDYRTN and DS16RTN (DS32RTN for 32 bit) signals, and must hold the address channel stable until these signals can be returned from the supported slaves. In some systems these timings can be most easily met by use of a default synchronous extended cycle for that master when active on the channel.

The above and other complexities can lead to expense. If the master is a processing unit, then memory, timers, interrupt controllers, and DMA function may also be on the shopping list. To offset the complexities, designers may wish to implement diagnostics and error checking/reporting as well.

A master is not passive like all other devices. It arbitrates for the channel, and upon gaining the grant, it takes over control of operations for one or more channel cycles. Because of this, the following positives are realized:

Random Operation and Bandwidth Reduction

The typical slave attachment depends on the processor of DMA controller for data transfer. Data must be exchanged with another entity as well as the slave attachment. For example, if a printer card is to be serviced as a DMA slave, the DMA controller will first read memory for the character and then write the character to the DMA printer. This is two operations. If the data were sourced from a communication port supported by DMA, then four operations would be involved, two with memory and one each with the communication port and the printer. If the slave devices being coupled together are slaves to the processor, with a complex interrupt-driven hardware sequence required to move data, then the overhead is seriously increased.

If one of the two participants is a master, then that master controls the slave device directly, executes the interrupt service from private memory (by supporting the interrupt on the channel directly), and only transfers data on the channel intended for the slave. Because of the preceding reasons, bandwidth and processor burden can be significantly reduced while improving the performance of the attachment. Indeed, when the interrupt is serviced by private memory, all the instructions required to support that interrupt are removed from the processor, and that interrupt is not burdened by other service duties of the processor. The results are more regular and insurance against an overrun prone slave. High Performance LAN, Communication and DASD intelligent subsystems are well suited to this type of attachment.

High-Level Functionality

When a master supports an interrupt, a processor is implied. When the processor is not busy servicing the supported channel interrupts, other duties can be performed. Background diagnostics, caching and/or encryption of data in stream, support of a high-level system interface, or intelligent support of a device interface may be possible.

Compatibility with a DMA Slave

If an attachment is first designed as a DMA slave and software evolves with an expectation of a relative priority within the system, the software structure will be subtly tied to the priority level. Overruns, limitation to performance, or an increase in cost to overcome these problems may result. Because the Micro Channel architecture uses the same arbitration channel for DMA and master arbitration, the priority does not have to change when a DMA slave is replaced by a master. Upgrades of attachments for increased functionality or performance need not face a compatibility problem due to program structure and can resolve other problems through BIOS or a high-level system interface.

Extension of System Performance

If a master operates on the channel as a processor, with its own private storage for execution, why not make it a processor? Indeed, if activities can be easily partitioned by rigid assignment of I/O to multiple processors or more dynamically through an advanced operating system, then multiple processors can peacefully coexist on the same channel. If I/O is a small fraction of channel activity and multiple processing units do not tax the channel excessively, one, two or three concurrent processors may reside together in the same system. System simulation may be required to verify a design point for a given configuration and/or application. Consider the following organization:

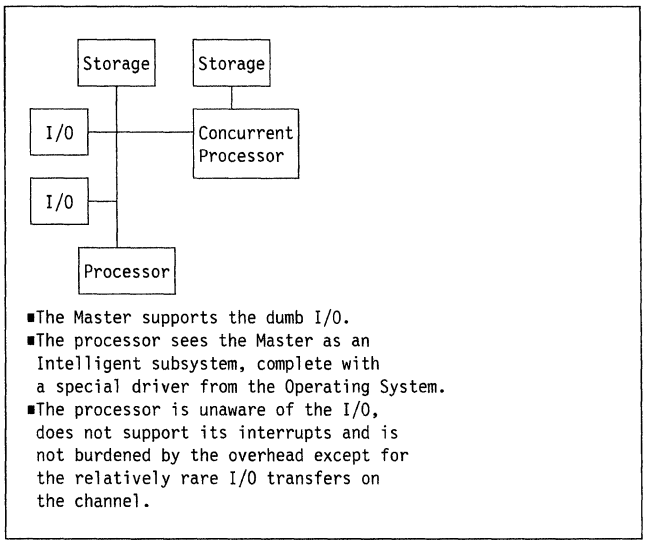

Storage
Storage
I/O
Concurrent Processor
I/O
Processor
▪The Master supports the dumb I/O.
▪The processor sees the Master as an Intelligent subsystem, complete with a special driver from the Operating System.
▪The processor is unaware of the I/O, does not support its interrupts and is not burdened by the overhead except for the relatively rare I/O transfers on the channel.

Figure 19. Concurrent I/O Processor Physical Organization

A number of intelligent subsystems can be configured from a collection of dumb I/O and concurrent processors!

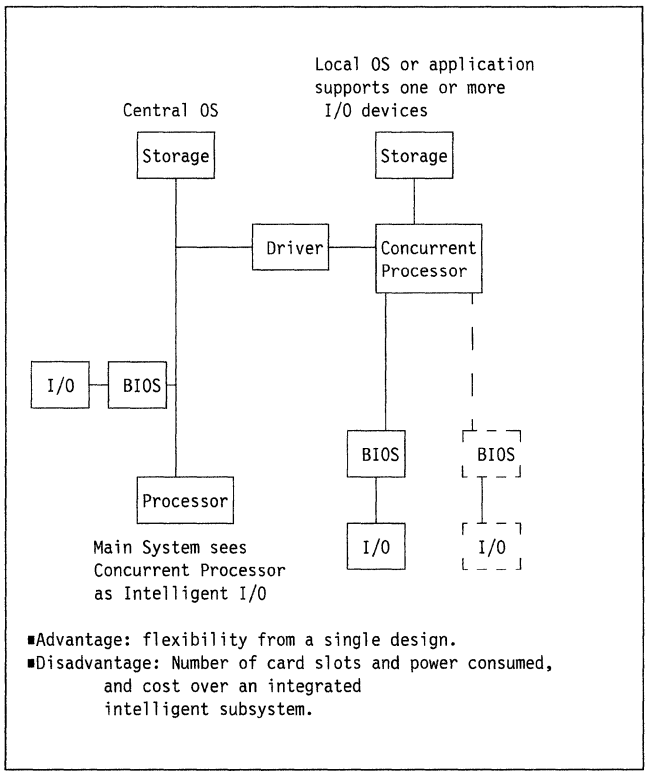


Figure 20. Concurrent I/O Processor Logical Organization

Micro Channel

Description

The Micro Channel is assembled from an address bus, a data bus, a transfer control bus, an arbitration bus, and multiple support signals. The channel architecture uses asynchronous protocols for control and data transfer between memory, I/O devices and the system microprocessor.

The following are characteristics of the Micro Channel.

- An I/O address width of 16 bits allows 8-, 16-, or 32-bit I/O transfers within a 64K range. A memory address width of 24 and 32 bits allow 8-, 16-, 24-, or 32-bit memory transfers within 16M- and 4G-byte ranges respectively. (K = 1024; M = 1,048,576; G = 1,073,741,824)
- Support of a central arbitration control point that allows up to 15 devices to arbitrate for control of the Micro Channel.
- A serial DMA protocol that supports eight DMA channels for 8- or 16-bit DMA transfers.
- Level sensitive interrupts with interrupt sharing on all levels.
- Programmable Option Select (POS) registers that replace hardware jumpers and switches. These registers allow high flexibility during system configuration at system power-on.
- Channel extension connectors that support the growth of additional channel features.
- Improved electro-magnetic compatibility
- Error reporting and recovery
- The Micro Channel Architecture does not support IBM Personal Computer adapters, and only supports adapters specifically designed for Micro Channel compatible systems.

Channel Definition

The channel provides all signal, power, and ground signals to the internal adapters.

The system board provides three types of channel connectors:

- 16-bit
- 16-bit with auxiliary video extension
- 32-bit.

The 16-bit channel has 77 signal lines, 29 power and ground lines, a separate audio ground line, five reserved lines, and four keyed positions in a dual 58-pin, 50-mil card edge connector. The 32-bit channel is an extended 16-bit channel designed to accommodate 32-bit addressing, 32-bit data transfers and Matched Memory cycles. The 32-bit extension provides an additional 31 signal lines, 15 power and ground lines and 16 reserved lines. The Matched Memory extension provides three signal lines, two ground lines, and three reserved lines. The 32-bit channel connector is a dual 93-pin, 50-mil card edge design. Every fourth pin on either side of each connector is at ac ground potential with side B offset from side A by 2 pins. This places each signal within 2.54 mm (0.1 inch) of a ground and minimizes current loop electromagnetic interference (EMI). The 50-mil channel connector reduces insertion force and matches surface mount technology line spacing. When designing adapters for the system, special design criteria must be considered.

The following is a diagram of the channel connectors.

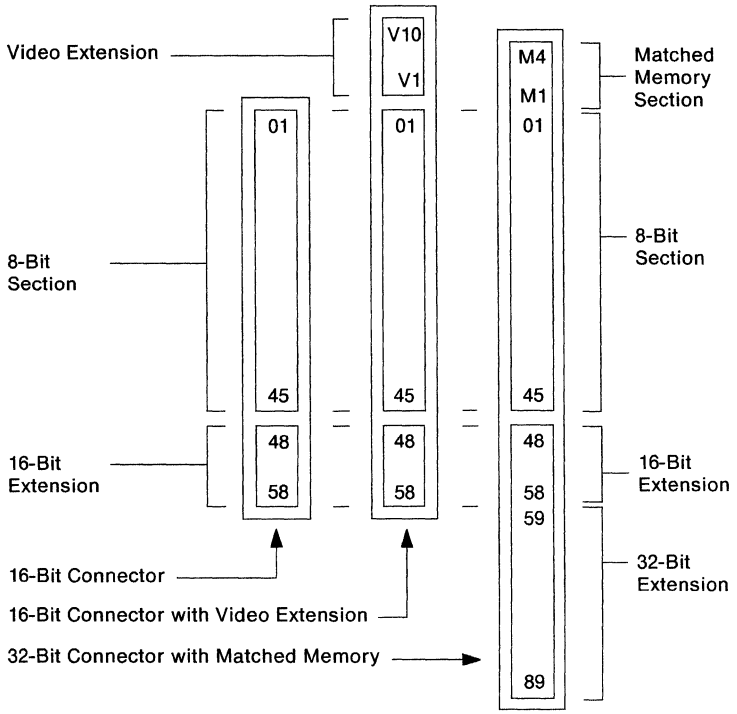


Figure 21. Micro Channel Connectors

Warning: Any signals shown or described as "Reserved" are not to be driven or received. These signals are reserved to allow compatibility with future implementations of the channel interface. Serious compatibility problems, loss of data, or permanent damage can result to features or the system if these signals are misused.

Signal Descriptions (16-Bit)

All of the logic signal lines are TTL-compatible. The following are the signals available on the channel.

A0 - A23: Address Bits 0 through 23: These lines are used to address memory and I/O slaves attached to the channel. A0 is the least-significant bit (LSB) while A23 is the most-significant bit (MSB). These 24 address lines allow access of up to 16 megabytes of memory. Only the lower 16 address lines (A0 - A15) are used for I/O operations and all 16 lines must be decoded by the I/O slave. A0 through A23 are generated by the system microprocessor. Valid addresses generated by the system microprocessor are unlatched on the channel and, if required, must be latched by the slaves using either the trailing edge of the '-address decode latch' (-ADL) signal or the leading edge of the 'command' (-CMD) signal. A0 through A23 must be driven with tri-state drivers.

D0 - D15: Data Bits 0 through 15: These lines provide data bus bits 0 to 7 (low byte) and 8 to 15 (high byte) for the system microprocessor and slaves. D0 is the LSB and D15 is the MSB. All 8-bit slaves on the channel must use D0 through D7 for communication with the system microprocessor. During read cycles, data is valid on these lines after the leading edge but before the trailing edge of -CMD, and must remain valid until after the trailing edge of -CMD. However, during write cycles, data is valid throughout the period when -CMD is active. D0 through D15 must be driven with tri-state drivers.

-ADL: -Address Decode Latch: This line, driven by the system microprocessor, is provided as a convenient mechanism for the slave to latch valid addresses and status bits. To ensure proper operation, slaves must use transparent latches to latch addresses and status bits with the trailing edge of -ADL or the leading edge of -CMD. -ADL is not active during Matched Memory cycles. -ADL is driven with a tri-state driver.

-CD DS 16 (n): -Card Data Size 16: This line is driven by 16-bit and 32-bit memory, I/O, or DMA slaves to provide an indication on the channel of a 16-bit or 32-bit data port at the location addressed. The signal is unlatched and derived as a valid address decode. All system logic receives this signal to support communication with 16-bit slaves. -CD DS 16 is not driven by an 8-bit memory, I/O, or DMA slave. The (n) indicates this signal line is unique to each channel

connector (one independent signal line per connector). -CD DS 16 is driven with a totem-pole driver.

-DS 16 RTN: -Data Size 16 Return: This output signal is a negative OR of the -CD DS 16 signals from each channel connector. If any device drives its -CD DS 16 active then this output is active. This signal is provided to allow channel resident bus masters (a device that arbitrates for, and controls the channel) to monitor the data size information. -DS 16 RTN must be driven with a bus driver.

-SBHE: -System Byte High Enable: This line indicates and enables transfer of data on the high byte of the data bus (D8 - D15), and is used with A0 to distinguish between high byte (D8 - D15) and low byte (D0 - D7) transfers. All 16-bit slaves decode this line but 8-bit slaves do not. -SBHE is driven with a tri-state driver.

MADE 24: Memory Address Enable 24: This line indicates when an extended address is used on the bus. If a memory cycle is in progress, MADE 24 inactive indicates an extended address space greater than 16M is being presented; MADE 24 active indicates an unextended address space less than or equal to 16M is being presented. This line is driven by the system microprocessor and decoded by all memory slaves, regardless of their address space size.

M/-IO: Memory/-Input Output: This signal distinguishes a memory cycle from an I/O cycle. When this signal is high, it indicates a memory cycle is in progress. When M/-IO is low, it indicates that an I/O cycle is in progress. M/-IO is driven with a tri-state driver.

-S0, -S1: -Status Bits 0 and 1: These lines indicate the start of a channel cycle and also define the type of channel cycle. When used with M/-IO, memory read/write operations are distinguished from I/O read/write operations. When M/-IO, -S0, and -S1 are all high, they do not signify a status cycle. These signals are latched by the slave, as required, using the leading edge of -CMD or the trailing edge of -ADL. -S0 and -S1 are driven with a tri-state driver.

Decoded commands are generated with latched status lines (-S0 and -S1), M/-IO, and -CMD.

Slaves must support a full decode of -S0 and -S1. The following figure shows the proper states of M/-IO, -S0, and -S1 in decoding I/O and memory read/write commands.

M/-IO	-S0	-S1	Function
0	0	0	Reserved A
0	0	1	IO Write
0	1	0	IO Read
0	1	1	Reserved B
1	0	0	Reserved C
1	0	1	Memory Write
1	1	0	Memory Read
1	1	1	Reserved D

Figure 22. I/O and Memory Transfer Controls

A decoded I/O Write command instructs an I/O slave to store the data on the data bus. The data must be valid on the bus from the leading edge of -CMD and must be held on the bus until after -CMD goes inactive. Addresses on the bus must be valid prior to -S0 going active.

A decoded I/O Read command instructs an I/O slave to drive its data onto the data bus. The data must be placed on the bus following the leading edge of -CMD, must be valid before the trailing edge of -CMD and must be held on the bus until -CMD goes inactive. Addresses on the bus must be valid prior to -S1 going active.

A decoded memory Write command instructs the memory to read the data on the data bus. The data must be valid on the bus from the leading edge of -CMD and must be held on the bus until after -CMD goes inactive. Addresses on the bus must be valid prior to -S0 going active.

A decoded memory Read command instructs the memory to drive its data onto the data bus. The data must be placed on the bus following the leading edge of -CMD. The data must be valid before the trailing edge of -CMD, and must be held on the bus until -CMD goes inactive. Addresses on the bus must be valid prior to -S1 going active.

-CMD: -Command: This signal is used to define when data is valid on the data bus. The trailing edge of this signal indicates the end of the bus cycle. This signal provides to the slave the indication of the duration during which the data is valid on the bus. During write operations, the data is valid on the bus throughout the period that -CMD is active. During read operations, the data is valid on the bus following the leading edge but before the trailing edge of -CMD and must be held on the bus until after the -CMD goes inactive. This signal can be used by the slaves to latch the address on the bus. Latched status lines gated by -CMD provide the timing control of valid data. To ensure proper operation slaves must use transparent latches to latch address and status

information with the leading edge of -CMD or the trailing edge of -ADL. -CMD is not active during Matched Memory cycles. -CMD must be driven with a tri-state driver.

-CD SFDBK (n): -Card Selected Feedback: When a memory slave or an I/O slave is addressed by the system microprocessor, this signal is driven active by the addressed slave, as a positive acknowledgment of its presence at the address specified. This signal can be used during diagnostics and installation to detect address space conflicts and/or defective field replaceable units or customer replaceable units. This signal is unlatched and driven by all slaves with a valid select decode. This signal is driven by a slave selected by any select mechanism except -CD SETUP. The slave does not drive -CD SFDBK during the configuration cycle.

Note: Memory supporting diagnostic software must not drive -CD SFDBK during the diagnostic operation.

The (n) indicates this signal line is unique to each channel connector (one independent signal line per connector). -CD SFDBK is driven with a totem-pole driver.

CD CHRDY (n): Channel Ready: This line, normally active (ready), is pulled inactive (not ready) by a memory or I/O slave to allow additional time to complete a channel operation. When using this line during a read operation, a slave promises that data will be valid on the data bus within the time specified after releasing the line to a ready state. The slave also holds the data for a sufficient amount of time for the system microprocessor to sample. A slave may also use this line during a write operation if more time is needed to store the data from the bus. The maximum time that CD CHRDY may be held inactive by a slave will not exceed 3.5 microseconds. The (n) indicates this signal line is unique to each channel connector (one independent signal line per connector). This signal is derived with a valid address decode ANDed with Status. CD CHRDY is driven with a totem-pole driver.

CHRDYRTN: Channel Ready Return: This output signal is a positive AND of the CD CHRDY signals from each channel connector. If all devices drive CD CHRDY active then this output is active. It is provided to allow channel resident masters to monitor the ready information. CHRDYRTN must be driven with a bus driver.

ARB0 - ARB3: Arbitration Bus Priority Levels: These lines comprise the arbitration bus and are used to present arbitrating bus participant priority levels. ARB0 through ARB3, the least-significant and most-significant bits respectively, support up to 16 priority levels for arbitration by the bus participants.

The highest hexadecimal value of the arbitration bus (hex F) has the lowest priority, and the lowest value (hex 0) has the highest priority. The arbitrating bus participant is allowed to change the state of the arbitration bus only immediately after the rising edge of ARB/-GNT. All arbitrating bus participants monitor the arbitration bus and the lower priority participants withdraw their priority levels by not activating less-significant arbitration bits.

The hexadecimal code of the highest priority requester is valid on the arbitration bus after a settling time. After the channel is granted, the highest priority participant continues to drive its priority lines. These bidirectional lines are active high and must be driven with open collector drivers.

ARB/-GNT: Arbitrate/-Grant: When high, this signal indicates that an arbitration cycle is in process. When low, this signal is the acknowledgement from the central arbitration control point (CACP) to the winning arbiter that channel control has been awarded. This signal is driven high by the CACP within a specified time after -S0, -S1, -BURST, and -CMD become inactive. The negative to positive transition of ARB/-GNT initiates an arbitration cycle and the positive to negative transition of ARB/-GNT terminates this arbitration cycle. Only the CACP activates and deactivates this line. This signal must be used by all arbitrating devices to gate their address data and transfer control bus drivers off during arbitration cycles. ARB/-GNT is driven with a bus driver.

-PREEMPT: -Preempt: This signal is driven by arbitrating bus participants (DMA slaves and intelligent bus controllers) to request usage of the channel through arbitration. Any arbitration bus participant with a channel request activates -PREEMPT and causes an arbitration cycle to occur. A requesting arbitration bus participant removes its preempt upon being granted the channel. This bidirectional line must be driven with an open collector driver.

-BURST: -Burst: This signal is driven by arbitrating bus participants to indicate to the CACP the extended use of the channel when transferring a block of data. This type of data transfer is referred to as a *burst cycle*. This line is shared by all arbitrating bus participants. This signal is driven active by the

arbitrating bus participant after being granted the channel. The participant must deactivate -BURST during or by the end of the last transfer cycle. -BURST must be driven with an open collector driver.

-TC: -Terminal Count: This line provides a pulse on the channel during a Read or Write command to indicate that the terminal count of the current DMA channel has been reached. This indicates to the DMA slave the last cycle to be performed of a pre-programmed DMA block transfer. -TC is available on the channel only during DMA operations. -TC is driven with a tri-state driver by the DMA Controller.

-IRQ 3 – 7, -IRQ 9 – 12, and -IRQ 14 – 15: -Interrupt Request: These lines are used to signal the system microprocessor that an I/O slave requires attention. They are prioritized with -IRQ 9 having the highest priority and -IRQ 7 having the lowest priority. The effective interrupt priority sequence is -IRQ (9-12, 14, 15, 3-7). An interrupt request is generated by the requesting interrupting slave driving one of the 'interrupt request' signals low. The polarity of 'interrupt request' signals makes it amenable for multiple slaves to share the same interrupt level. This is referred to as *interrupt sharing*. These lines must be driven with an open collector driver.

-CD SETUP (n): -Card Setup: This signal is driven by system board logic to individually select channel connectors during system configuration and error recovery procedures. When this signal is activated, a specific channel connector is selected and the configuration data space of the adapter is accessed. The card ID and the configuration data of the adapter can be obtained by an I/O read operation with -CD SETUP active. The configuration data is stored by the adapter as an I/O write operation with -CD SETUP active. Each slot has a unique -CD SETUP. The (n) indicates this signal line is unique to each channel connector (one independent signal line per connector). This line is driven with a totem-pole driver.

-CHCK: -Channel Check: This line is used to indicate a serious error (such as a parity error) which threatens continued operation of the system. -CHCK is driven active to indicate the error condition and must remain low until the -CHCK interrupt handler resets it. -CHCK is driven with an open collector driver to allow sharing.

AUDIO: Audio Sum Node: This line on the channel is an audio voltage sum node. This line is used to drive audio signals from an adapter to the system audio output or to transfer audio signals between adapters. The frequency response of the audio line is 50 Hz to 10 kHz $\pm$ 3 dB. The maximum signal amplitude is 2.5 volts peak to peak, at a dc offset of 0 $\pm$ 50 millivolts. The noise level is limited to a maximum of 50 millivolts peak to peak.

AUDIO GND: Audio Ground: This is a separate ground return for the audio system.

OSC: Oscillator: This line is a high speed clock with a frequency of 14.31818 MHz $\pm$ 0.01%. The high level (more than 2.3 volts) pulse width and the low level (less than 0.8 volt) pulse width must not be less than 20 nanoseconds each.

CHRESET: Channel Reset: This signal is generated by the system board logic to reset or initialize all adapters upon power on or during a low line voltage condition. During a power-on sequence, CHRESET is active for a specified minimum time. In addition, the system can activate this signal under program control. CHRESET is driven with a bus driver.

-REFRESH: -Refresh: This line is driven by the system board logic and is used to indicate that a memory refresh operation is in progress. While this line is active, a memory read operation occurs. The address lines contain the memory locations being refreshed. Nine lines, A0 - A8, are activated. -REFRESH timing may be inconsistent and must not be used as a timing mechanism.

Signal Descriptions (32-Bit)

A24 – A31: Address Bits 24 through 31: These lines are used with A0 through A23 to address memory attached to the channel. A0 is the LSB and A31 is the MSB. These 32 address lines allow access of up to 4G of memory. Only the lower 16 address lines (A0 through A15) are used for I/O operations. A24 through A31 are generated by the system microprocessor. Valid addresses generated by the system microprocessor are unlatched on the bus and, if required, must be latched by the slaves using either the trailing edge of -ADL or the leading edge of -CMD. A0 through A31 are driven with tri-state drivers.

-BE0 – -BE3: -Byte Enable 0 through 3: These lines are used during data transfers involving 32-bit slaves to indicate which data bytes will be placed on the bus. Data transfers of 8-, 16-, 24-, or 32-contiguous bits are controlled by -BE0 through -BE3 during transfers involving 32-bit slaves only. These lines are driven by the system microprocessor or other master when TR 32 is inactive, and by the Central Translator logic (for those operations involving a 16-bit master with a 32-bit slave) when TR 32 is active.

These lines are unlatched on the bus and, if required, must be latched by 32-bit slaves. -BE0 through -BE3 are driven with tri-state drivers.

D16 – D31: Data Bits 16 through 31: These lines are used with D0 through D15 to provide data bus bits to the system microprocessor, intelligent bus controller, and slaves. D0 is the LSB and D31 is the MSB. All 32-bit transfers from the bus master to 8-bit slaves are converted to four 8-bit transfers, all transmitted on lines D0 through D7. During read cycles, data is valid on these lines after the leading edge of -CMD but before the trailing edge of -CMD. However, during write cycles, data is valid throughout the period when -CMD is active. D0 through D31 must be driven with tri-state drivers.

-CD DS 32 (n): -Card Data Size 32: This line is driven by 32-bit memory, I/O, or DMA to provide an indication on the bus of a 32-bit data port at the location addressed. -CD DS 32(n) is unlatched and derived from a valid address decode. All 32-bit slaves must drive this signal. -CD DS 32(n) is inactive for an 8- or 16-bit data port of 32-bit slaves. The (n) indicates this signal line is specific to a channel connector position (one independent signal per connector). -CD DS 32(n) must be driven with a totem-pole driver.

Note: The DMA controller used in this system does not support 32-bit DMA slaves.

-DS 32 RTN: -Data Size 32 Return: This output signal is a negative OR of the -CD DS 32 signals from each channel connector. If any device drives its -CD DS 32 active then this output is active. This signal is provided to allow channel resident masters to monitor the data size information. -DS 32 RTN is be driven with a bus driver.

TR 32: Translate 32: This line is driven inactive by 32-bit masters and received by the Central Translator Logic. TR 32 can also be received by any 32-bit slave. When TR 32 is inactive, the 32-bit master drives BE0 through BE3. When TR 32 is active, the Central Translator Logic drives BE0 through BE3. TR 32 must be driven by a tri-state driver.

Signal Descriptions (Matched Memory)

The following are descriptions of the Matched Memory signals.

-MMC: -Matched Memory Cycle: This signal is driven by the system board logic to indicate to the bus channel slaves that the system microprocessor is controlling the bus and is able to run a Matched Memory bus cycle. -MMC is driven by a tri-state driver.

-MMCR: -Matched Memory Cycle Request: This is a bus cycle control input signal. -MMCR is driven by a 16- or 32-bit channel slave to request the faster cycle available on the system bus. -MMCR is driven active from an address decode and M/-IO. This signal is wired separately for three channel connectors on the system board and ORed by the system logic. The 80386 microprocessor is the only controlling device that can run Matched Memory cycles. If -MMCR is asserted active by an 8-bit channel slave, or by a 16- or 32-bit channel slave during a DMA bus cycle or other nonmicroprocessor bus cycle, a Basic Transfer cycle is run and -MMCR is not honored. CD CHRDY is used to extend the Matched Memory cycle if required. -ADL and -CMD remain inactive for the entire Matched Memory bus cycle.

-MMC CMD: -Matched Memory Cycle Command: This signal is an output to the bus and is generated for the 80386 microprocessor bus cycles only. -MMC CMD is used to define when data is valid on the bus during a Matched Memory bus cycle. The trailing edge of the signal indicates the end of the Matched Memory bus cycle. As with -CMD, this signal provides the slave with an indication of the duration during which the data is valid on the bus, and when to either latch the data from the bus or stop driving the data onto the bus.

Matched Memory Cycle

This section describes the Model 80 80386 *Matched Memory cycle* bus operation supported by the Micro Channel. The Model 80 implementation is such that a channel slave can be either a memory or I/O slave with a data bus width of 16 or 32 bits. No 8-bit devices are allowed to run Matched Memory cycles.

The Matched Memory cycle is a system-unique function supported by the Micro Channel, and is provided for memory subsystem expansion. This function allows the most efficient data transfer capability between the 80386 and channel memory. The system board ROM and RAM (held on the 80386 local bus) and the 32-bit memory expansion adapter (held on the channel) adhere to the Matched Memory cycle protocol described in this section.

The Model 80 supports two types of 80386 bus cycles on the channel; Matched Memory bus cycles and Basic Transfer bus cycles. Matched Memory bus cycles are a minimum of 3 (16 MHz) clocks or 187-nanosecond cycle time. Basic Transfer bus

cycles are a minimum of 4 (16 MHz) clocks or 250-nanoseconds cycle time. The target channel slave must request the Matched Memory cycle on a bus-cycle by bus-cycle basis, through the -MMCR (Matched Memory Cycle Request) signal or it will get the default Basic Transfer cycle.

An 80386 bus cycle begins when the 'address', 'byte enables', and bus definition signals become valid in the first clock cycle of the bus cycle. Driving either of the status bits active signals the start of the channel bus cycle. The status state (Ts) is the first clock cycle. After Ts, the command state Tc is entered. Channel slaves respond to the bus operation during Tc, either transferring read data to or accepting write data from the system bus. Tc states may be repeated as often as necessary to assure sufficient time for the channel slave to respond. CD CHRDY is used to determine whether Tc is repeated. A repeated Tc state is called a 62.5-nanosecond wait state.

If -MMCR is asserted active by the channel slave, then -MMC CMD is driven by the system microprocessor.

If -MMCR is not returned from the channel slave, a Basic Transfer bus cycle is run. Basic Transfer and Matched Memory bus cycles can be mixed in any manner because the mechanism is completely dynamic on a bus cycle basis. Bus cycles can be extended until CD CHRDY is sampled as active.

CD CHRDY is asserted inactive by a memory or channel slave to allow additional time to complete a Basic Transfer or Matched Memory bus cycle when the default cycle length is not sufficient to service that device. See "Matched Memory Cycle Timing (No Waits)" on page 44 for the timing diagrams related to the 80386 Matched Memory cycle.

Signal Descriptions (Auxiliary Video Extension)

The following are signal descriptions for the Auxiliary Video extension of the channel connector.

VSYNC: This signal is the vertical sync signal used to drive the display. See also the ESYNC signal description.

HSYNC: This signal is the horizontal sync signal used to drive the display. See also the ESYNC signal description.

BLANK: This signal is connected to the BLANK input of the video Digital-to-Analog Converter (DAC). When active (0 volts), this signal tells the DAC to drive its analog color outputs to 0 volts. See also the ESYNC signal description.

P7 - P0: These eight signals contain digital video information and comprise the picture element (PEL) address inputs to the video DAC. See also the EVIDEO signal description.

DCLK: This signal is the video PEL clock that is used by the video DAC to latch the digital video signals, P7 through P0. P7 through P0 are latched on the rising edge of DCLK inside the DAC.

This signal is also connected to the EXTCLK input of the Video Graphics Array (VGA), and may be driven by the auxiliary video connector and used as the input clock to the VGA.

Note: When this configuration is used, the VGA may not be the source of the digital video signals presented to the DAC (P7 - P0); rather, P7 through P0 must be driven from the auxiliary video connector. See also the EDCLK signal description.

ESYNC: This signal is the output enable signal for the buffer that drives the BLANK, VSYNC, and HSYNC signals. ESYNC is tied to 5V through a pull-up resistor so that an open circuit on the ESYNC pin produces 5V.

When ESYNC equals 5V, BLANK, VSYNC, and HSYNC are sourced from the VGA BLANK, VSYNC, and HSYNC outputs respectively. When ESYNC equals 0V, BLANK, VSYNC, and HSYNC are driven from the auxiliary video connector.

EVIDEO: This signal is the output enable signal for the buffer that drives P7 through P0. EVIDEO is tied to 5V through a pull-up resistor so that an open circuit on the EVIDEO pin produces 5V.

When the EVIDEO signal equals 5V, P7 through P0 are sourced from the VGA outputs, P7 through P0. When the EVIDEO signal equals 0V, P7 through P0 are driven from the auxiliary video connector.

EDCLK: This signal is the output enable signal for the buffer that drives DCLK. EDCLK is tied to 5V through a pull-up resistor so that an open circuit on the EDCLK pin produces 5V.

When EDCLK equals 5V, DCLK is sourced from the VGA DCLK output, and is received by the auxiliary video connector and DAC. The VGA Miscellaneous Output

register should be configured so that it is not selecting clock source 2 when EDCLK equals 5V.

When EDCLK equals 0V, DCLK is driven from the auxiliary video connector to the EXTCLK input of the VGA chip and to the DAC.

Note: When this configuration is used, the VGA may not be the source of the digital video signals presented to the DAC (P7 - P0); rather, P7 through P0 must be driven from the auxiliary video connector. The Miscellaneous Output register must be programmed to select clock source 2.

Micro Channel Connector (16-Bit)

The 16-bit Micro Channel connector is organized into:

- An 8-bit section
- A 16-bit extension.

A key is provided between the 8-bit section and 16-bit extension for mechanical alignment.

Note: The auxiliary video extension, 32-bit extension, and Matched Memory extension are shown separately beginning on page 24.

The following figures show the signals and voltages assigned to the 16-bit channel connector.

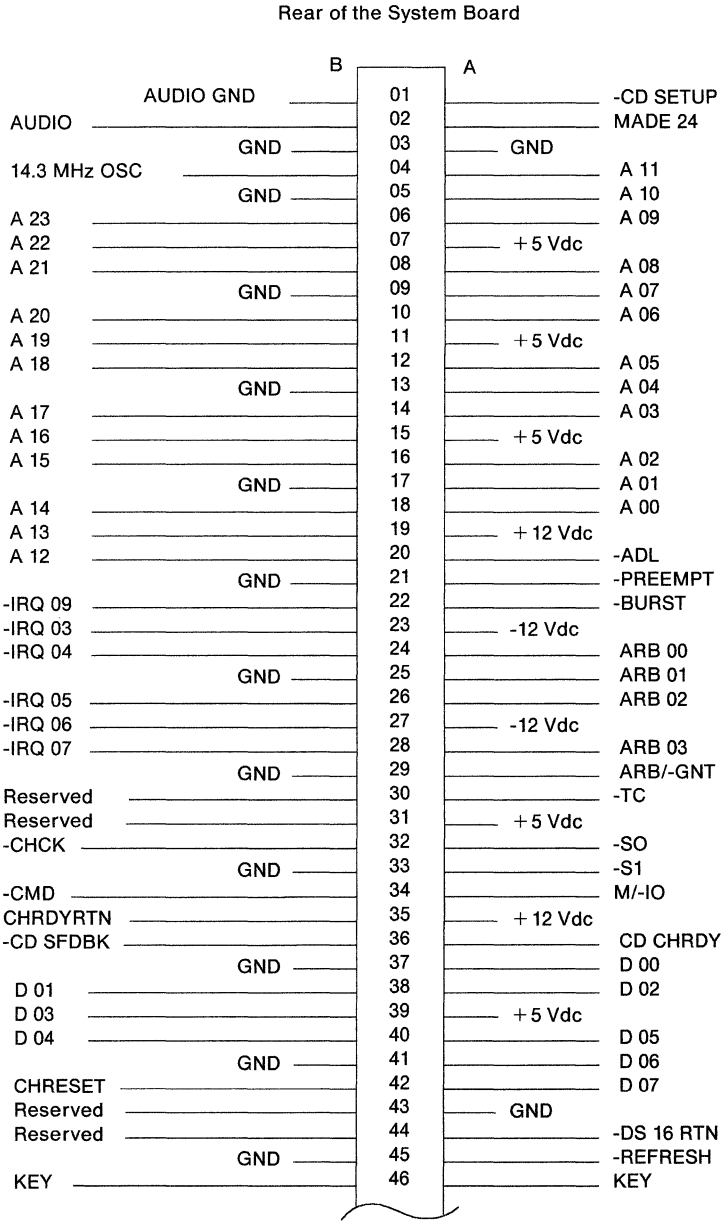


Figure 23. Channel Connector Voltage and Signal Assignments (8-Bit Section)

The following are the signals and voltages assigned to the channel connector 16-bit extension.

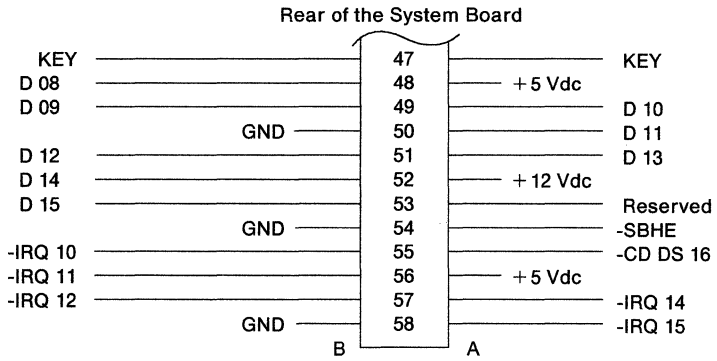


Figure 24. Micro Channel Connector 16-Bit Extension Voltage and Signal Assignments

Micro Channel Connector (Auxiliary Video Extension)

This connector extends the 16-bit Micro Channel connector to accommodate video adapters that interface with the system board video subsystem.

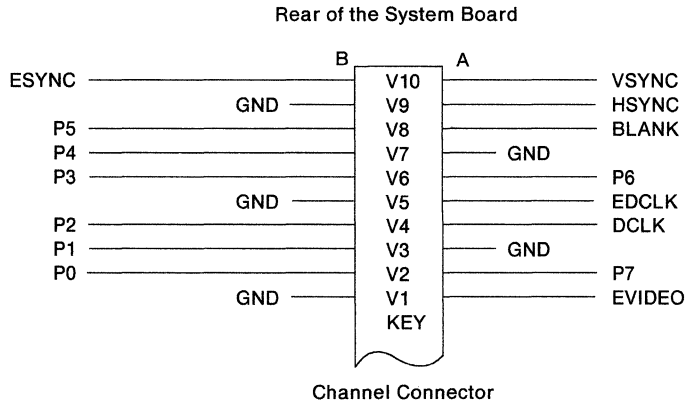


Figure 25. Auxiliary Video Connector

Micro Channel Connector (32-Bit Extension)

This connector extends the 16-bit Micro Channel connector to accommodate 32-bit addressing and 32-bit data transfers.

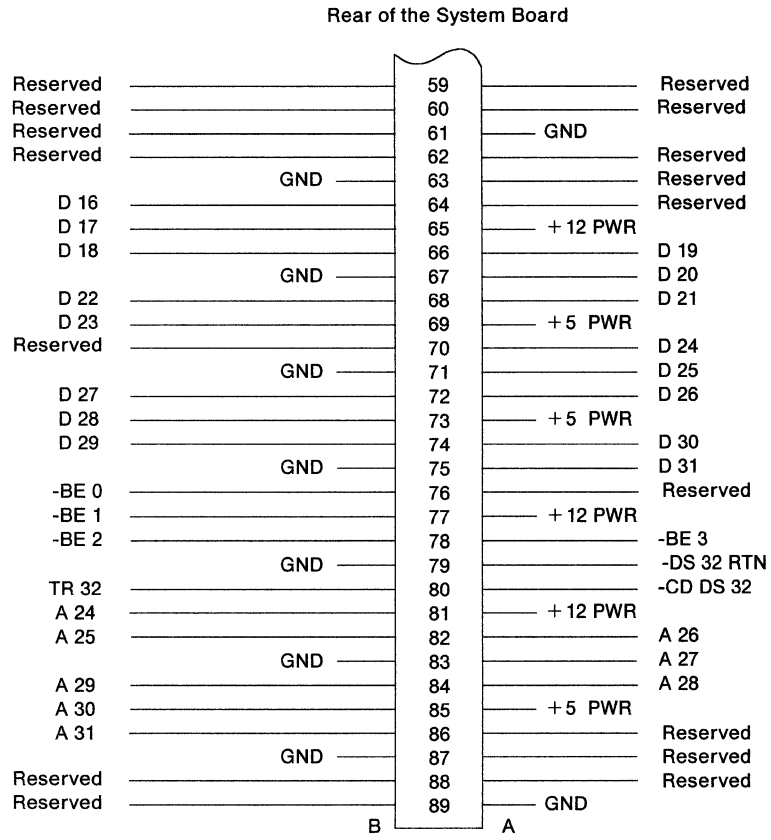


Figure 26. Micro Channel Connector 32-Bit Extension Voltage and Signal Assignments

Micro Channel Connector (Matched Memory Extension)

This connector extends the 32-bit Micro Channel connector to accommodate Matched Memory cycles.

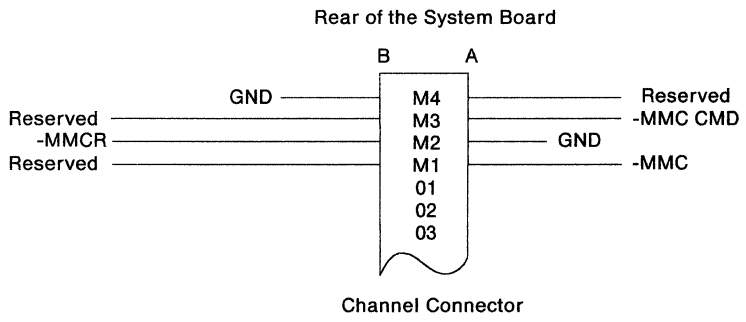


Figure 27. Micro Channel Connector Matched Memory Extension Voltage and Signal Assignments

Channel Signal Groups (16-Bit, 32-Bit, and Matched Memory)

The following figure lists 16-bit, 32-bit, and Matched Memory Micro Channel signals and shows which type of driver or receiver is required to be compatible with the system board.

Signal Name	Sys Logic	DMA Cntlr	Intelligent Bus Cntlr	DMA Slave	MEM Slave	I/O Slave	Driver Type
	D/R	D/R	D/R	D/R	D/R	D/R	(Signal Group)
A(0-23)	D/-	D/-	D/-	-/R	-/R	-/R	TS (1)
D(0-15)	D/R	D/R	D/R	D/R	D/R	D/R	TS (2)
-ADL	D/-	D/-	D/-	-/O	-/O	-/O	TS (1)
-CD DS 16 (n)	-/R	-/R	-/R	#/-	#/-	#/-	TP (3)
-DS 16 RTN	D/-	-/R	-/R	-/-	-/-	-/-	BD (4)
-SBHE	D/-	D/-	D/-	-/#	-/#	-/#	TS (1)
MADE 24	D/-	D/-	D/-	-/-	-/R	-/-	TS (1)
M/-IO	D/-	D/-	D/-	-/R	-/R	-/R	TS (1)
-S0,-S1	D/-	D/-	D/-	-/R	-/R	-/R	TS (1)
-CMD	D/-	D/-	D/-	-/R	-/R	-/R	TS (1)
-CD SFDBK (n)	-/R	-/-	-/-	D/-	D/-	D/-	TP (3)
CD CHRDY (n)	O/R	-/-	-/-	O/-	O/-	O/-	TP (3)
CHRDYRTN	D/-	-/R	-/R	-/-	-/-	-/-	BD (4)
ARB(0-3)	O/R	-/R	D/R	D/R	-/-	-/-	OC (5)
-BURST	O/R	D/R	D/-	#/-	-/-	-/-	OC (5)
-PREEMPT	O/R	-/-	D/R	D/#	-/-	-/-	OC (5)
ARB/-GNT	D/-	-/R	-/R	-/R	-/-	-/-	BD (4)
-TC	-/-	D/-	-/-	-/O	-/-	-/-	TS (1)
-IRQ (*)	-/R	O/-	O/-	O/-	-/-	O/-	OC (6)
-CD SETUP (n)	D/-	-/#	-/R	-/R	-/R	-/R	TP (7)
-CHCK	-/R	#/-	D/O	D/-	D/-	D/-	OC (6)
-REFRESH	D/R	-/O	-/O	-/O	-/R	-/O	TS (1)
OSC	D/-	-/O	-/O	-/O	-/O	-/O	CD (7)
CH RESET	D/-	-/O	-/R	-/R	-/R	-/R	BD (4)
A(24-31)	D/-	D/-	D/-	-/R	-/R	-/R	TS (1)
-BE(0-3)	D/-	D/-	D/-	-/R	-/R	-/R	TS (1)
D(16-31)	D/R	D/R	D/R	D/R	D/R	D/R	TS (2)
-CD DS 32 (n)	-/R	-/R	-/R	#/-	#/-	#/-	TP (3)
-DS 32 RTN	D/-	-/-	-/O	-/-	-/-	-/-	BD (4)
TR 32 RTN	-/-	-/-	-/O	-/-	O/-	O/-	TS (1)
-MMC	D/-	-/-	-/-	-/-	-/O	-/O	TS (1)
-MMCR	-/R	-/-	-/-	-/-	O/-	O/-	TP(3)
-MMC CMD	D/-	-/-	-/-	-/-	-/O	-/O	TS (1)

KEY

D = Drive Enabled
O = Optional
R = Receive Enabled
- = Not Implemented
# = Some are Required
OC = Open Collector
TS = Tri-State
TP = Totem-Pole
BD = Bus Driver
CD = Clock Driver

* IRQ (9-12, 14, 15, 3-7)

Figure 28. Driver/Receiver Requirements and Options

The following notes apply to the driver and receiver options listed in Figure 28.

Notes:

1. During the Reset, an active CHRESET must degate all bus drivers.
2. During the Reset, the state of all signals is unknown.
3. A CD SETUP (n) line is driven to only one channel connector at a time.
4. All pull-up resistors are provided by the system logic and pulled up to +5 volts.
5. Loading Current: 1.6 milliamps maximum load per channel connector except signal group 5. The loading current of group 5 is 1.0 milliamp maximum per channel connector.
6. Loading Capacitance:
   - 15-pf maximum capacitance loading permitted for adapter for OSC and Group 5 signals (see Figure 29 for Group 5 definition) and 20 pf maximum capacitance loading permitted for adapter for all other signals. (The value refers to the capacitance from the adapter side of the connector to the adapter driver/receiver.)
   - Total capacitance seen by the driver is 200 pf maximum for Group 5 signals and OSC and 240 pf maximum for all other signals. The symbol (#) refers to average capacitance across 0.1-volt to 2.3-volt interval.
7. An Open Collector can be either an open collector device or a tri-state device wired with the input grounded and using the 'enable' line to control the output.
8. The electromagnetic interference (EMI) potential of a bus driver increases as its output voltage transition time decreases. Therefore, the drivers with output transitions greater than 1 volt per nanosecond should be used only when essential to meet channel timing requirements. The figure lists the role of driver or receiver during a given operation being performed. The names of the physical packaging of the logic should not be confused with the performed functions.

Signal Group	Driver Type
1, 2	Tri-state with 24 ma sinking capacity.
3	Totem-pole (TP) with current sinking capacity of 6 ma.
4	Bus Driver (BD) with current sinking capacity of 24 ma.
5, 6	Open Collector (OC) with 24 ma sinking capacity.
7	Unique drivers: Totem-pole (TP) or Tri-state (TS) with current sinking capacity of 6 ma. Clock Driver (CD) with 24 ma sinking capacity.

Figure 29. Signal Driver Types

Channel Signal Groups (Auxiliary Video Extension)

An adapter using the auxiliary video connector must not exceed the loading limits shown below for any auxiliary video connector signal pin it is receiving:

- C max = 15.0 pF
- I_{IL} min = - 1.6 mA
- I_{IH} max = 50.0 uA.

An adapter using the auxiliary video connector must meet the following minimum requirements for any auxiliary video connector signal pin it is driving:

- C min = 150.0 pF
- I_{OL} min = 10.0 mA - VSYNC and HSYNC
  = 2.0 mA - All other signals
- I_{OH} max = - 4.0 mA - VSYNC and HSYNC
  = - 0.25 mA - All other signals.

Micro Channel Critical Timing Parameters

This section provides timing diagrams for the Micro Channel. All timings are related to a nominal cycle. Systems and adapters may alter this nominal cycle through various mechanisms. Developers should use care that hardware and software designs operate over the ranges specified and do not depend on a given performance level. Portability may be adversely affected by time-dependent hardware and software.

Basic Transfer Cycle

This section provides the specification for critical timing parameters for the Basic Transfer cycle.

Simplified Basic Transfer Cycle

Most microprocessor and DMA operations transfer data with the same control sequence. For transfers other than Matched Memory transfers, the signals appear on the channel in the following sequence:

1. Address bus, MADE 24, M/-IO, and -REFRESH (if applicable) become valid, beginning the cycle.
2. The 'status' signals become valid.
3. The 'address decode latch' (-ADL) signal becomes valid.
4. In response to an unlatched address decode, MADE 24, and M/-IO, the adapter returns:
   - -CD SFDBK
   - -CD DS 16 (if the attachment is capable of 16-bit operation).
   - -CD DS 16 and -CD DS 32 (if the attachment is capable of 32-bit operation).
5. In response to an unlatched address decode, MADE 24, M/-IO, and Status, the adapter drives CH RDY inactive if the cycle is to be extended.
6. Write data appears on the bus (for the Write cycle).
7. -CMD becomes active and -ADL become inactive.
8. The 'status' signals become inactive.
9. The 'address' signals become invalid in preparation for the next cycle.
10. In response to an address change:
    - -CD SFDBK is set inactive by the attachment.
    - -CD DS 16 is set inactive by the attachment.
    - -CD DS 32 is set inactive by the attachment.
11. If CD CHRDY has been set inactive, the system holds in this state indefinitely until CD CHRDY is set active. This line should not be held inactive longer than 3.0 microseconds.
12. The attachment places Read data on the bus in preparation for the trailing edge of -CMD (for the Read cycle).
13. The address, 'status' signals, and M/-IO for the next cycle may become valid.
14. -CMD goes inactive, ending the cycle.

Note: The address and Status can be overlapped with the preceding cycle to minimize the memory access time impact on performance.

The sequence is as follows:

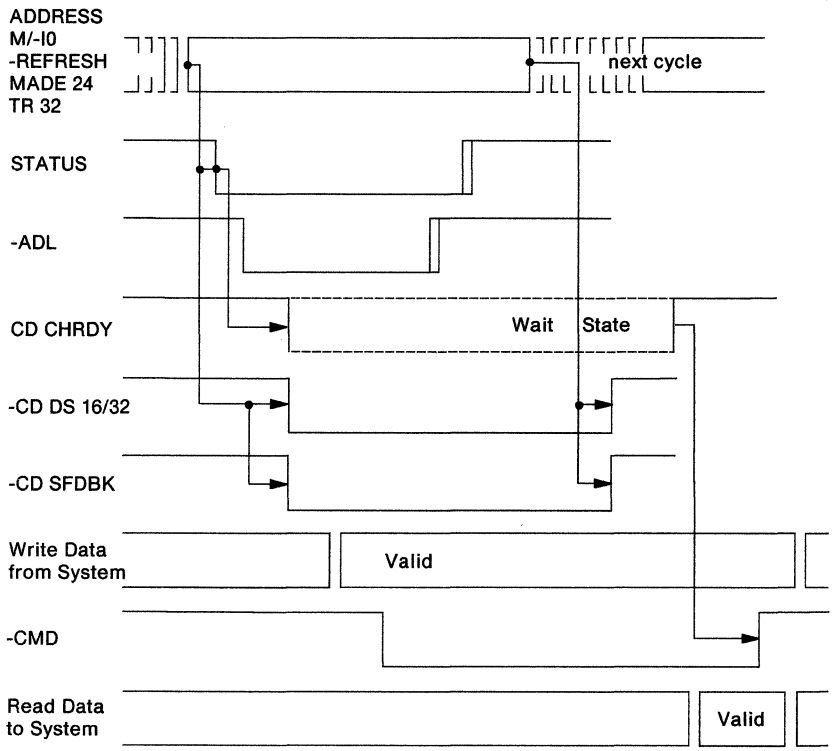


Figure 30. Overview of the Basic Transfer Cycle

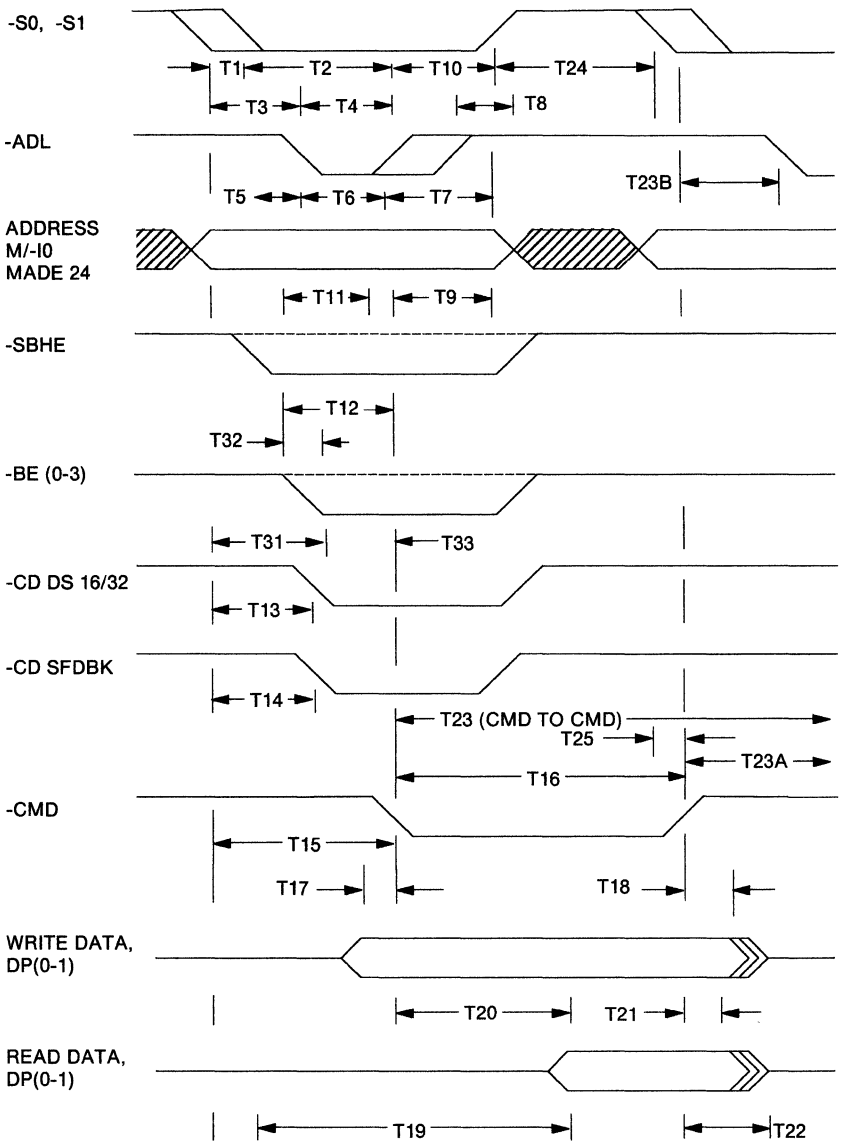
I/O and Memory Cycle

The basic I/O and memory cycle timing diagrams are shown on the following pages. Matched Memory cycle timing diagrams begin on page 44. The basic I/O and memory cycle timing diagrams appear in the order listed below.

- Default cycle (200 nanoseconds minimum)
- Synchronous Extended cycle (300 nanoseconds minimum) - Special case
- Asynchronous Extended cycle (≥300 nanoseconds minimum) - General case.

A Synchronous or an Asynchronous Extended cycle is caused by a slave depending upon its use of CD CHRDY. In general, a slave releases CD CHRDY asynchronously causing an Asynchronous Extended cycle.

Default Cycle



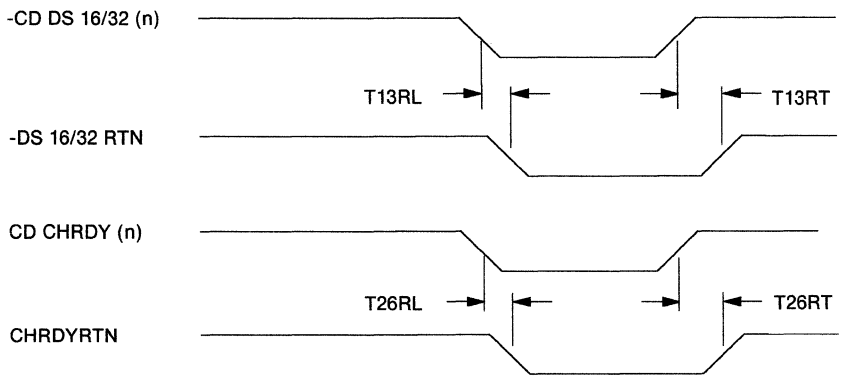
	Timing Parameter	Min/Max	Note
T1	Status active (low) from ADDRESS,M/-IO,-REFRESH valid	10 / - ns	
T2	-CMD active (low) from Status active (low)	55 / - ns	2
T3	-ADL active (low) from ADDRESS,M/IO,-REFRESH valid	45 / - ns	
T4	-ADL active (low) to -CMD active (low)	40 / - ns	
T5	-ADL active (low) from Status active (low)	12 / - ns	
T6	-ADL pulse width	40 / - ns	
T7	Status hold from -ADL inactive (high)	25 / - ns	2
T8	ADDRESS,M/-IO,-REFRESH,-SBHE hold from -ADL inactive	25 / - ns	2
T9	ADDRESS,M/-IO,-REFRESH,-SBHE hold from -CMD active (low)	30 / - ns	3
T10	Status hold from -CMD active	30 / - ns	2
T11	-SBHE setup to -ADL inactive	40 / - ns	2
T12	-SBHE setup to -CMD active	40 / - ns	2
T13	-CD DS 16/32 active (n) (low) from ADDRESS,M/-IO,-REFRESH valid	- / 55 ns	3
T14	-CD SFDBK active (low) from ADDRESS,M/-IO,-REFRESH valid	- / 60 ns	1
T15	-CMD active (low) from Address valid	85 / - ns	2
T16	-CMD pulse width	90 / - ns	
T17	Write data setup to -CMD active (low)	0 / - ns	
T18	Write data hold from -CMD inactive (high)	30 / - ns	
T19	Status to Read Data valid (Access Time)	- / 125 ns	
T20	Read data valid from -CMD active (low)	- / 60 ns	
T21	Read data hold from -CMD inactive (high)	0 / - ns	
T22	Read data bus tri-state from -CMD inactive (high)	- / 40 ns	
T23	-CMD active to next -CMD active	190 / - ns	4
T23A	-CMD inactive to next -CMD active	80 / - ns	
T23B	-CMD inactive to next -ADL active	40 / - ns	
T24	Next Status active (low) from Status inactive	30 / - ns	
T25	Next Status active (low) to -CMD inactive	- / 20 ns	
T31	-BE(0-3) active from Address valid (32-bit masters only)	- / 40 ns	
T32	-BE(0-3) active from -SBHE, A0, A1 active	- / 30 ns	
T33	-BE(0-3) active to -CMD active	10 / - ns	

Figure 31. I/O and Memory Default Cycle (200 nanoseconds minimum)

Notes:

1. All slaves must drive -CD SFDBK whenever selected either by the system microprocessor or the DMA Controller. The slaves do not drive -CD SFDBK when they are selected by the 'setup' signal.
2. To ensure proper operation, slaves must use transparent latches to latch addresses and status bits with the trailing edge of -ADL or with the leading edge of -CMD.
3. -CD DS 16/32 and -CD SFDBK must be driven by *unlatched address decodes* because the next address may come early into the current cycle.
4. Any master in any system, including the system microprocessor or DMA controller, can operate at a performance level less than the level specified. Designers should not design to a given performance level, because this level can be reduced by CD CHRDY, a lower microprocessor rate, a lower DMA controller rate, or system contention.

Default Cycle Return Signals



	Timing Parameter	Min/Max	Note
T13RL	-CD DS 16/32 (n) low to -DS 16/32 RTN low	- / 20 ns	1
T13RT	-CD DS 16/32 (n) high to -DS 16/32 RTN high	- / 20 ns	1
T26RL	CD CHRDY (n) low to CHRDYRTN low	- / 20 ns	2
T26RT	CD CHRDY (n) high to CHRDYRTN high	- / 20 ns	3

Figure 32. Default Cycle Return Signals (200 nanoseconds minimum)

Notes:

1. This signal is developed from a negative OR of signals received from each channel connector.
2. This signal is developed from a positive AND of signals received from each channel connector. This is a change from other systems using the Micro Channel architecture.
3. This signal is developed from a positive AND of signals received from each channel connector.

Synchronous Special Case of Extended Cycle

A Synchronous Extended cycle occurs when a slave releases CD CHRDY synchronously within the specified time after the leading edge of -CMD. The slave provides the Read data within a specified time from -CMD. The timing sequence is illustrated by the following figure.

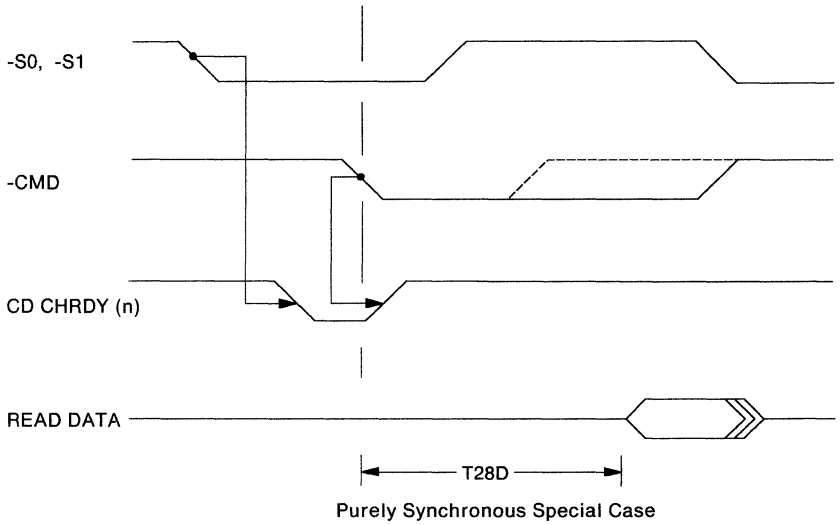
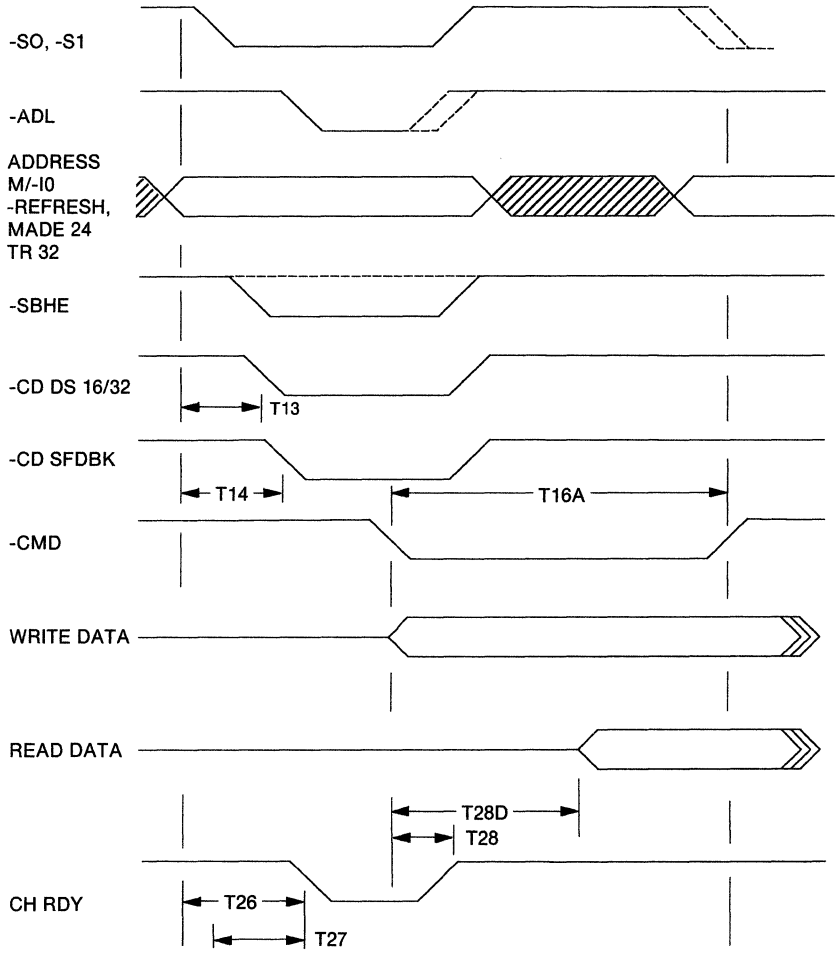


Figure 33. Timing Sequence for the Synchronous Special Case of Extended Cycle

Synchronous Extended Cycle (300 nanoseconds minimum - Special Case)



	Timing Parameter	Min/Max	Note
T13	-CD DS 16/32 (n) active (low) from ADDRESS,M/-IO,-REFRESH valid	- / 55 ns	2
T14	-CD SFDBK (n) active (low) ADDRESS,M/-IO,-REFRESH valid	- / 60 ns	2
T16A	-CMD pulse width	190 / - ns	
T26	CD CHRDY (n) inactive (low) from ADDRESS valid	- / 60 ns	3, See T27
T27	CD CHRDY (n) inactive (low) from Status active	0 / 30 ns	3
T28	CD CHRDY (n) release (high) from -CMD active (low)	0 / 30 ns	1
T28D	Read Data valid from -CMD active (when used along with T28)	0 / 160 ns	1

This figure shows only the parameters additional to the default cycle.
All other parameters are the same as the default cycle.

Figure 34. Synchronous Extended Cycle (300 nanoseconds minimum - Special Case)

Notes:

1. CD CHRDY is released by a slave performing a 300-nanosecond extended cycle synchronous with the leading edge of -CMD. Since CD CHRDY is generally an asynchronous signal, this is referred to as a purely synchronous special case.
2. This is the same as default cycle timing (listed here for emphasis).
3. T27 is valid only when Status becomes active 30 nanoseconds or more after the address is valid.
4. If Status overlaps with previous -CMD, then CD CHRDY state is not valid during the overlapped period.
5. Slaves must not hold CD CHRDY inactive (low) in excess of 3.0 microseconds.

Asynchronous Extended Cycle (General Case)

An Asynchronous Extended cycle occurs when a slave releases CD CHRDY asynchronously. However, the slave provides the Read data within the specified time from CD CHRDY release. The timing sequence is illustrated by the following figure.

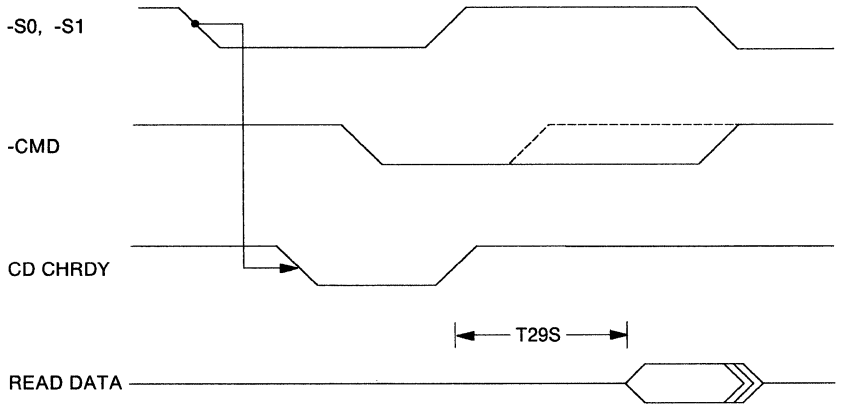
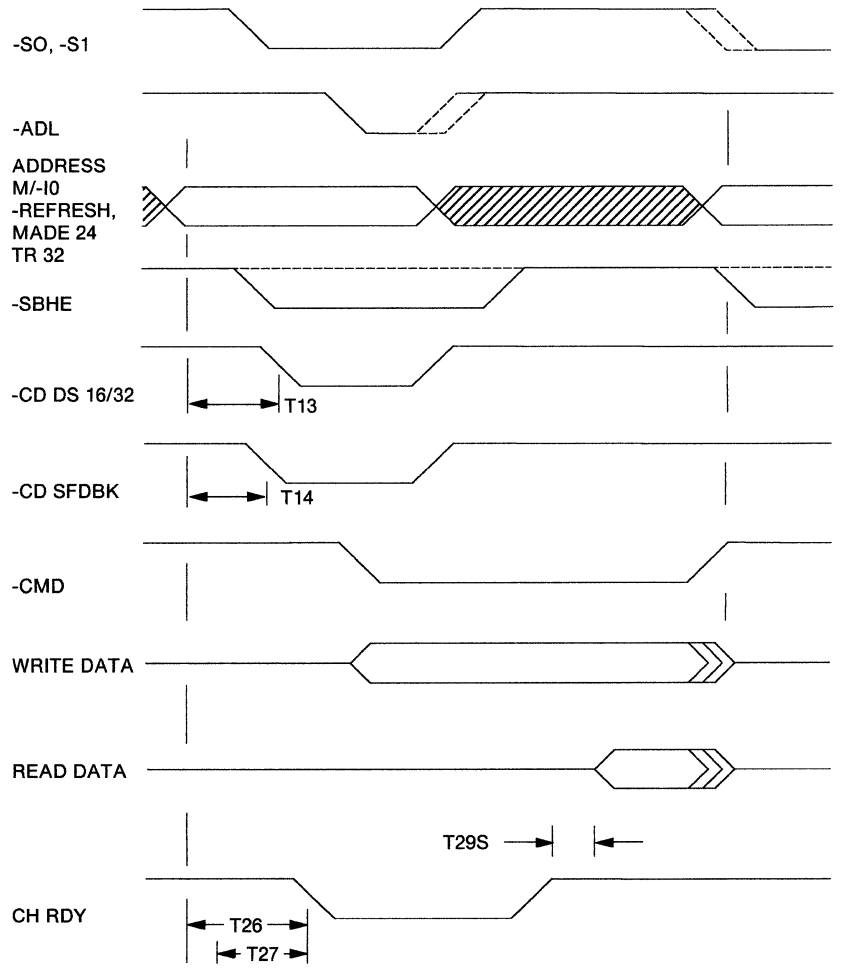


Figure 35. Timing Sequence for the Asynchronous Extended Cycle (General Case)

Asynchronous Extended Cycle (≥300 nanoseconds minimum - General Case)



	Timing Parameter	Min/Max	Note
T13	-CD DS 16/32 (n) active (low) from ADDRESS,M/-IO,-REFRESH valid	- / 55 ns	2
T14	-CD SFDBK (n) active (low) ADDRESS,M/-IO,-REFRESH valid	- / 60 ns	2
T26	CD CHRDY (n) inactive (low) from ADDRESS valid	- / 60 ns	See T27
T27	CD CHRDY (n) inactive (low) from Status active	0 / 30 ns	
T29S	Read data from slave valid from CD CHRDY (n) active (high)	- / 60 ns	1
This figure shows only the parameters additional to the default cycle. All other parameters are the same as the default cycle.			

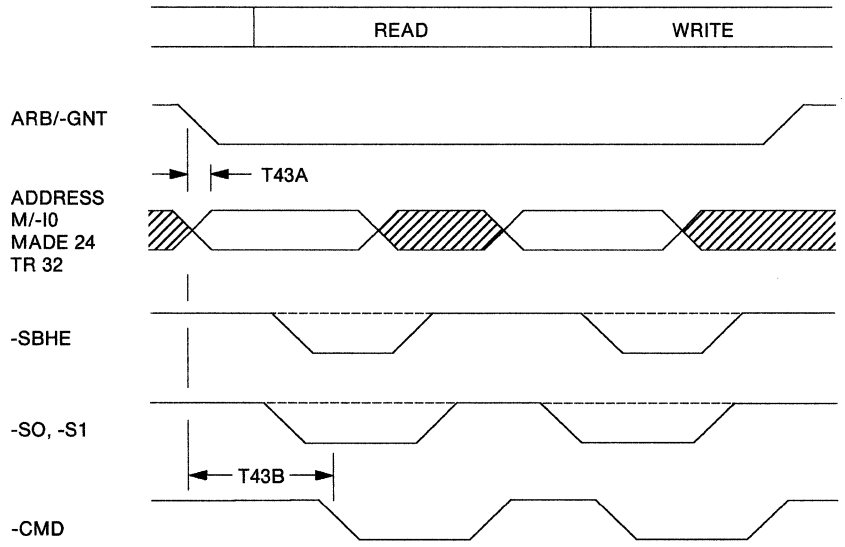
Figure 36. Asynchronous Extended Cycle (≥300 nanoseconds minimum - General Case)

Notes:

1. CD CHRDY is released asynchronously by a slave performing 300 nanoseconds minimum cycle. The slave must present the Read data within the time specified after the release of CD CHRDY.
2. This is the same as default cycle timing.
3. T27 is valid only when Status becomes active 30 nanoseconds or more after the address is valid.
4. If Status overlaps with the previous -CMD, then the CD CHRDY state is not valid during the overlapped period.
5. Slaves must not hold CD CHRDY inactive (low) in excess of 3.0 microseconds.

DMA Timing

First Cycle After Grant

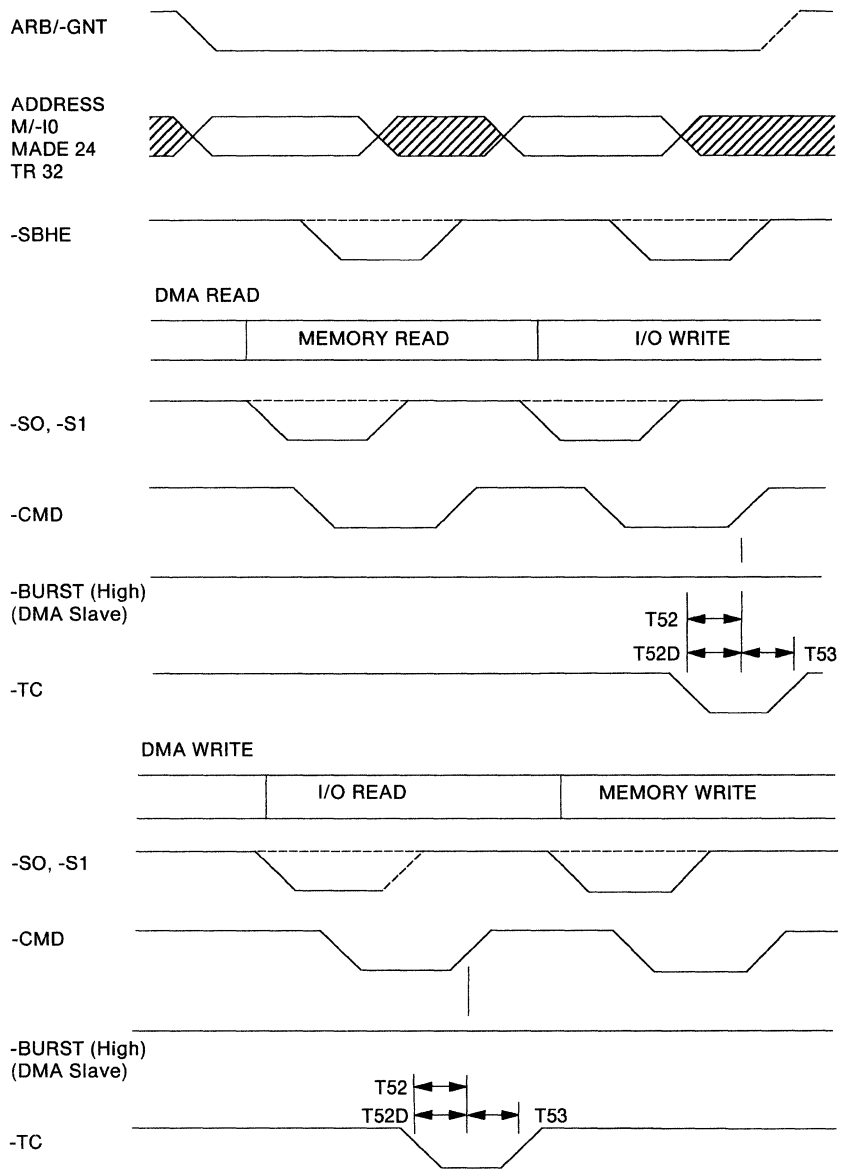


	Timing Parameter	Min/Max
T43A	ADDR valid from ARB/-GNT low	0 / - ns
T43B	-CMD active from ARB/-GNT low	115 / - ns

Figure 37. First Cycle After Grant

Note: After the grant, a controller must allow 30 nanoseconds for a slave to generate an internal acknowledgment that it has been selected. During the first cycle, the controller must additionally allow 30 nanoseconds before sampling -CD DS 16/32, -CD SFDBK, and CD CHRDY, if it places an address on the bus within 30 nanoseconds after grant active.

Single DMA Transfer (DMA Controller Controlled)



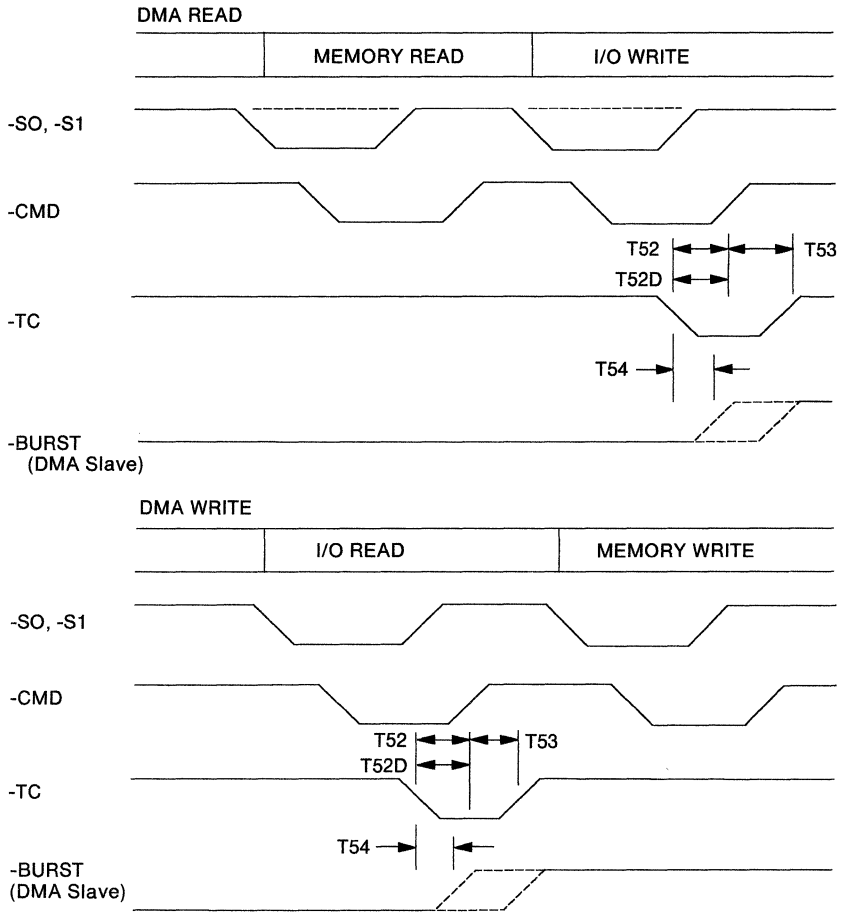
	Timing Parameter	Min/Max	Note
T52	-TC setup to -CMD inactive	30 / - ns	
T52D	-TC setup to -CMD inactive	15 / - ns	2
T53	-TC hold to -CMD inactive	10 / - ns	

Figure 38. Single DMA Transfer (DMA Controller Controlled)

Notes:

1. Only those timing parameters additional to those specified for the Basic Transfer cycle are included here.
2. Only for devices using a 200-nanosecond minimum default cycle.

Burst DMA Transfer (DMA Controller Terminated)



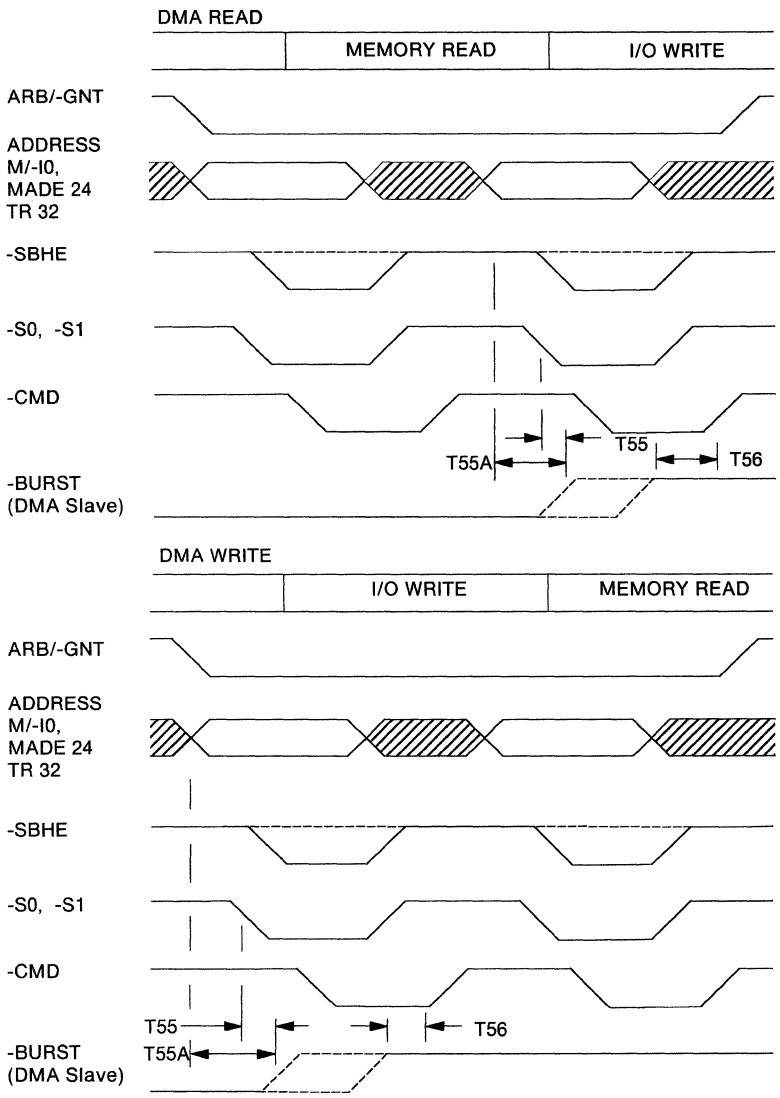
	Timing Parameter	Min/Max	Note
T52	-TC setup to -CMD inactive	30 / - ns	
T52D	-TC setup to -CMD inactive	15 / - ns	2
T53	-TC hold from -CMD inactive	10 / - ns	
T54	-BURST released by the DMA slave from -TC active	- / 30 ns	

Figure 39. Burst DMA Transfer (DMA Controller Terminated)

Notes:

1. Only those timing parameters additional to those specified for the Basic Transfer cycle are included here.
2. Only for devices using a 200-nanosecond minimum default cycle.

Burst DMA Transfer (DMA Slave Terminated - Default Cycle 200 nanoseconds)



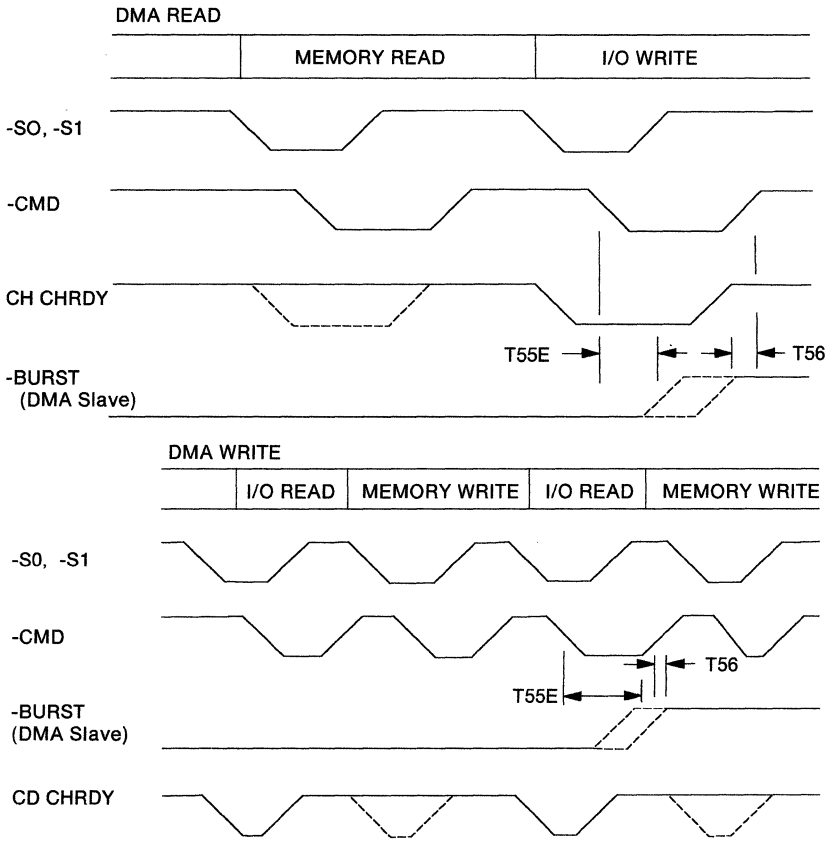
	Timing Parameter	Min/Max	Note
T55	-BURST released by the DMA slave from the last I/O Status active (Default cycle only)	- / 40 ns	2
T55A	-BURST released by the DMA slave from the last I/O ADDRESS valid (Default cycle only)	- / 70 ns	2
T56	-BURST inactive (high) setup to -CMD inactive	35 / - ns	3

Figure 40. Burst DMA Transfer (DMA Slave Terminated - Default Cycle 200 nanoseconds)

Notes:

1. Only those timing parameters additional to those specified for the Basic Transfer cycle are included here.
2. After releasing -BURST and on receiving -SBHE, if the DMA slave has another cycle to perform, it must redrive -BURST.
3. -BURST (high) setup time (T56) to the end of -CMD must be guaranteed during the last I/O Write cycle to prevent the DMA Controller from starting the next cycle. This setup time (T56) is guaranteed by the sum of -BURST release (T55/T55A) by the DMA slave and the -BURST resistor capacitor (RC) restoration time. The RC restoration time must not exceed 70 nanoseconds. Note that T56 is the same for Default and Extended cycles.

Burst DMA Transfer (DMA Slave Terminated - Synchronous Extended Cycle 300 nanoseconds)



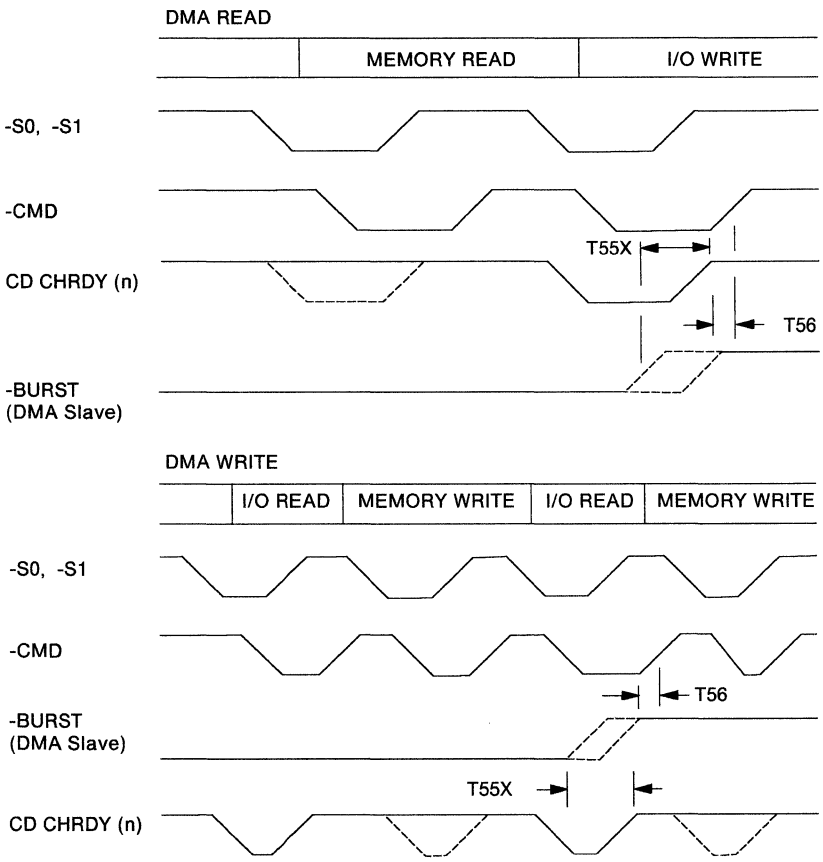
	Timing Parameter	Min/Max	Note
T55E	-BURST released by the DMA slave from the last -CMD active (Extended cycles only)	- / 80 ns	
T56	-BURST inactive (high) setup to -CMD inactive	35 / - ns	2

Figure 41. Burst DMA Transfer (DMA Slave Terminated - Synchronous Extended Cycle 300 nanoseconds)

Notes:

1. Only those timing parameters additional to those specified for the Basic Transfer cycle are included here.
2. -BURST (high) setup time (T56) to the end of -CMD must be guaranteed during the last I/O Write cycle to prevent the DMA Controller from starting the next cycle. This setup time (T56) is guaranteed by the sum of -BURST release (T55E) by the DMA slave and the -BURST RC restoration time. The RC restoration time must not exceed 70 nanoseconds. Note that T56 is the same for Default and Extended cycles.

Burst DMA Transfer (DMA Slave Terminated - Asynchronous Extended Cycle ≥300 nanoseconds)



	Timing Parameter	Min/Max	Note
T55X	-BURST released by the DMA slave before CD CHRDY (n) active (high) (Async. Extended cycles only)	50 / - ns	
T56	-BURST inactive (high) setup to -CMD inactive	35 / - ns	2

Figure 42. Burst DMA Transfer (DMA Slave Terminated - Asynchronous Extended Cycle ≥300 nanoseconds)

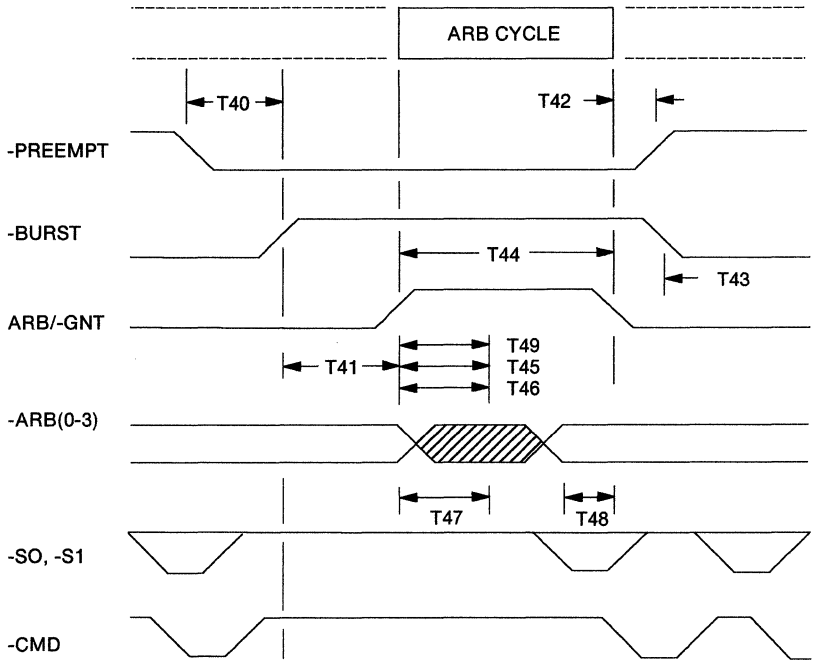
Notes:

1. Only those timing parameters additional to those specified for the Basic Transfer cycle are included here.
2. -BURST (high) setup time (T56) to the end of -CMD must be guaranteed during the last I/O Write cycle to prevent the DMA Controller from starting the next cycle. This setup time (T56) is guaranteed by the sum of -BURST release (T55X) by the DMA slave and the -BURST RC restoration time. The RC restoration time must not exceed 70 nanoseconds. Note that T56 is the same for Default and Extended cycles.

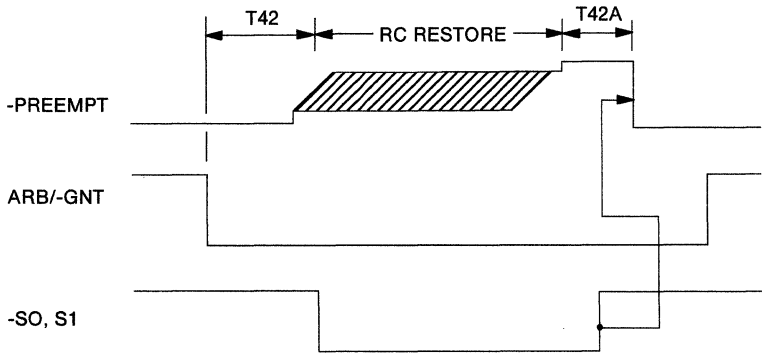
Arbitration Timing

This section provides the specification for critical timing parameters for arbitration protocol.

Arbitration Cycle



Exiting Inactive State



	Timing Parameter	Min/Max	Note
T40	-PREEMPT active (low) to End of Transfer	0 / 7.8 µs	1
T41	ARB/-GNT high from End of Transfer	30 / - ns	6
T42	-PREEMPT inactive (high) from ARB/-GNT low	0 / 50 ns	
T42A	-PREEMPT inactive (high) to Status inactive (Exiting Inactive State)	20 / - ns	5
T43	-BURST active (low) from ARB/-GNT low (By Bursting DMA slave)	- / 50 ns	4
T44	ARB/-GNT high	300 / - ns	2
T45	Driver turn-on delay from ARB/-GNT high	0 / 50 ns	3
T45A	Driver turn-on delay from lower priority line	0 / 50 ns	3
T46	Driver turn-off delay from ARB/-GNT high	0 / 50 ns	3
T47	Driver turn-off delay from higher priority line	0 / 50 ns	3
T48	Arbitration bus stable before ARB/-GNT low	10 / - ns	
T49	Tri-state drivers from ARB/-GNT high	- / 50 ns	

Figure 43. Arbitration Cycle

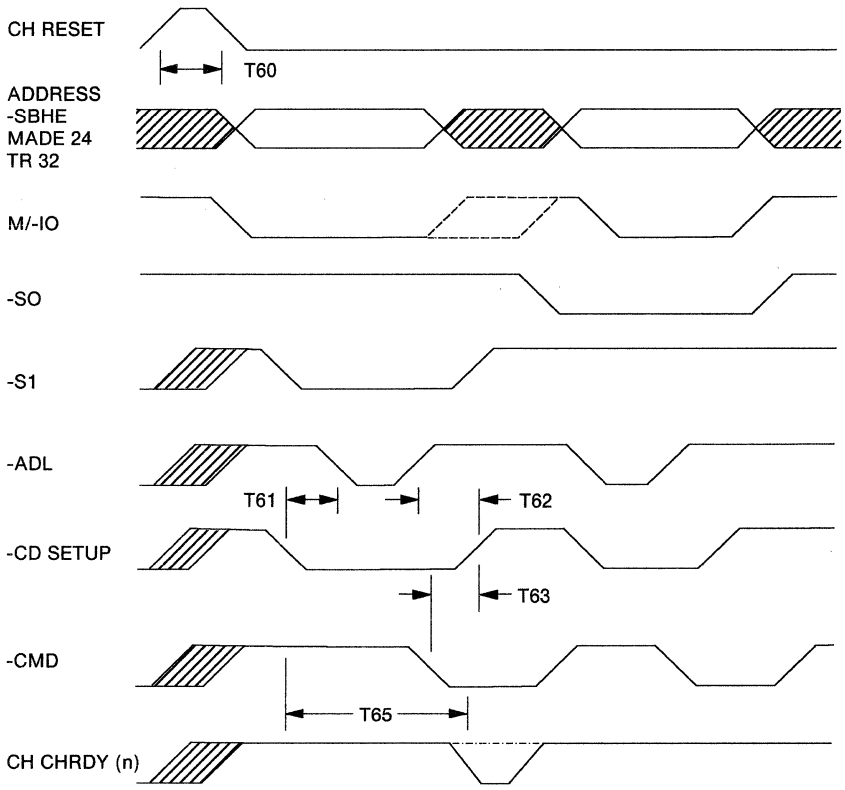
Notes:

1. The intent of this parameter is to limit the maximum non-preemptive ownership of the bus.
2. The value shown applies only to the special case implementation involving the central arbitration control point, and is provided for pulse width and portability considerations only. Arbitration can be extended by refresh or error recovery procedures. An arbiter should decode a win of the grant by a combined decode of the arbitration bus and the ARB/-GNT. The minimum arbitration time can be 100 nanoseconds when a level 0 arbiter and the central arbitration control point coordinate. In this special case, a central arbitration control point can terminate arbitration prematurely at 100 nanoseconds.
3. T45, T45A, T46, and T47 must be satisfied by ARB (0-3) drivers of all arbitrating bus participants.
4. This parameter is applicable to all bus winners.
5. This represents the timing requirement after the RC line delay. This window is available for devices to detect inactive -PREEMPT and exit Inactive State.
6. Because no maximum is specified, a channel attached Bus Master must degate bus drivers at the End of Transfer condition. The End of Transfer condition must be held stable until arbitration begins.

Configuration Timing

This section provides the specification for critical timing parameters for the system configuration protocol.

Setup Cycle



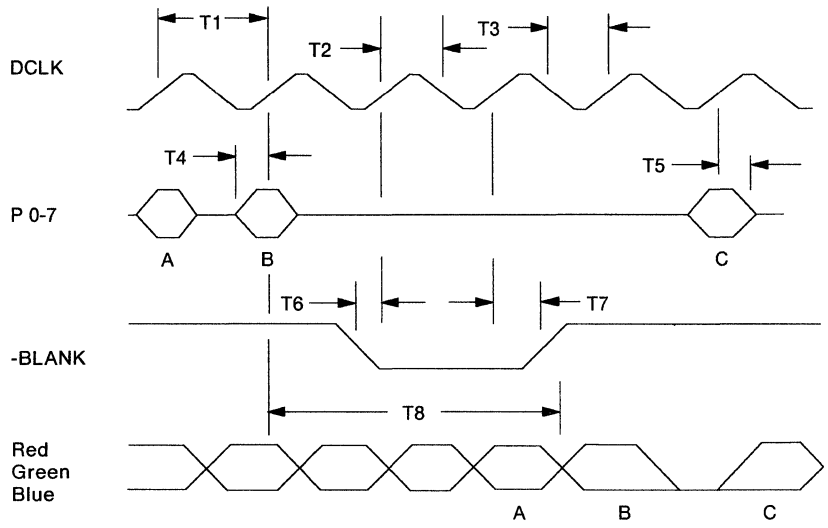
	Timing Parameter	Min/Max	Note
T60	CHRESET active (high) pulse width	100 / - ms	
T61	-CD SETUP (n) active (low) to -ADL active (low)	15 / - ns	
T62	-CD SETUP (n) hold from -ADL inactive (high)	25 / - ns	
T63	-CD SETUP (n) hold from -CMD active (low)	30 / - ns	
T65	CD CHRDY (n) inactive (low) from -CD SETUP (n) active	- / 100 ns	3

Figure 44. Setup Cycle

Notes:

1. Only those timing parameters that are different or additional to those specified for the Basic Transfer cycle are included here.
2. The Setup cycle is 300 nanoseconds minimum (default). A valid non-adapter selecting address must be present on the bus during system configuration.
3. A slave is allowed to extend the Setup cycle beyond 300 nanoseconds using CD CHRDY. The slave qualifies the leading edge of CD CHRDY with active Status.
4. Setup cycles are restricted to 8-bit transfers.

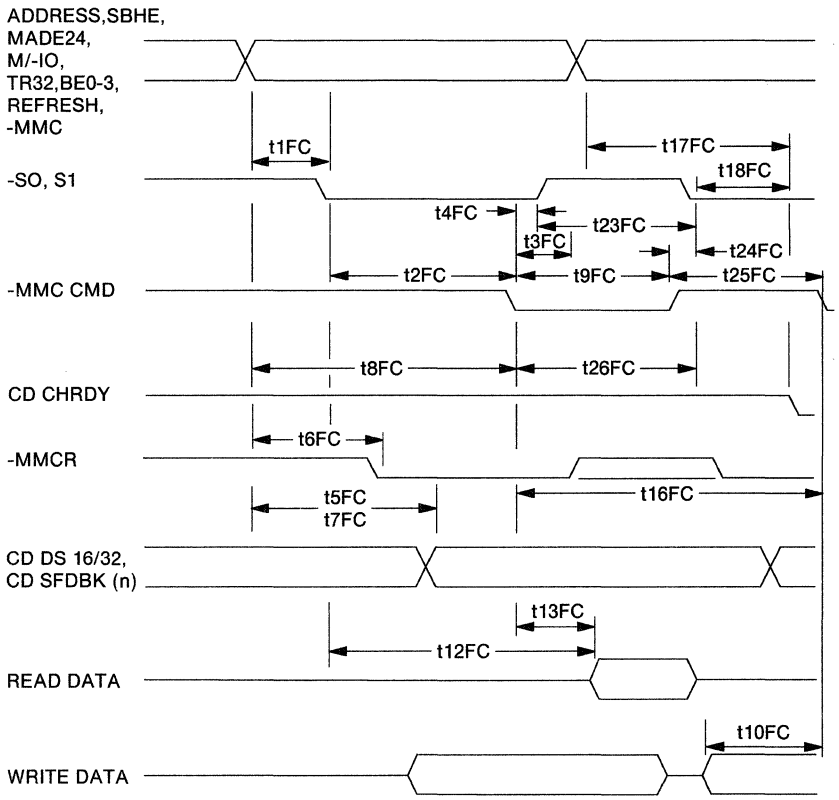
Auxiliary Video Connector Timing



	Timing Parameter	Min/Max	Note
T1	PEL Clock Period (tclk)	28	10,000
T2	Clock Pulse Width High (tch)	7	10,000
T3	Clock Pulse Width Low (tcl)	9	10,000
T4	PEL Set-up Time (tps)	4	-
T5	PEL Hold Time (tph)	4	-
T6	Blank Set-up Time (tbs)	4	-
T7	Blank Hold Time (tbh)	4	-
T8	Analog Output Delay (taod)	3(T1) + 5	3(T1) + 30

Figure 45. Auxiliary Video Connector Timing (DAC Signals)

Matched Memory Cycle Timing (No Waits)



Symbol	Description	Min/Max
t1 FC	ADDR valid to Status active	10 / - ns
t2 FC	Status valid to -MMC CMD active	82 / - ns
t3 FC	ADDR hold from -MMC CMD active	20 / - ns
t4 FC	Status hold from -MMC CMD active	25 / - ns
t5 FC	CD DS 16/32 active from ADDR valid	- / 55 ns
t6 FC	-MMCR active from ADDR valid	- / 55 ns
t7 FC	-CD SFDBK active from ADDR valid	- / 60 ns
t8 FC	ADDR valid to -MMC CMD active	100 / - ns
t9 FC	-MMC CMD pulse width	85 / - ns
t10 FC	Write data valid to -MMC CMD active	0 / - ns
t11 FC	Write data hold from -MMC CMD inactive	30 / - ns
t12 FC	Read data valid from Status active	- /145 ns
t13 FC	Read data valid from -MMC CMD active	- / 60 ns
t14 FC	Read data hold from -MMC CMD inactive	0 / - ns
t15 FC	Read data off delay from -MMC CMD inactive	- / 40 ns
t16 FC	-MMC CMD active to next -MMC CMD active	180 / - ns
t17 FC	CD CHRDY valid from ADDR valid	- / 70 ns
t18 FC	CD CHRDY valid from Status active	- / 30 ns
t23 FC	Status inactive pulse width	30 / - ns
t24 FC	-MMC CMD inactive to Status active	- / 5 ns
t25 FC	-MMC CMD inactive pulse width	85 / - ns
t26 FC	-MMC CMD active to Status active	90 / - ns

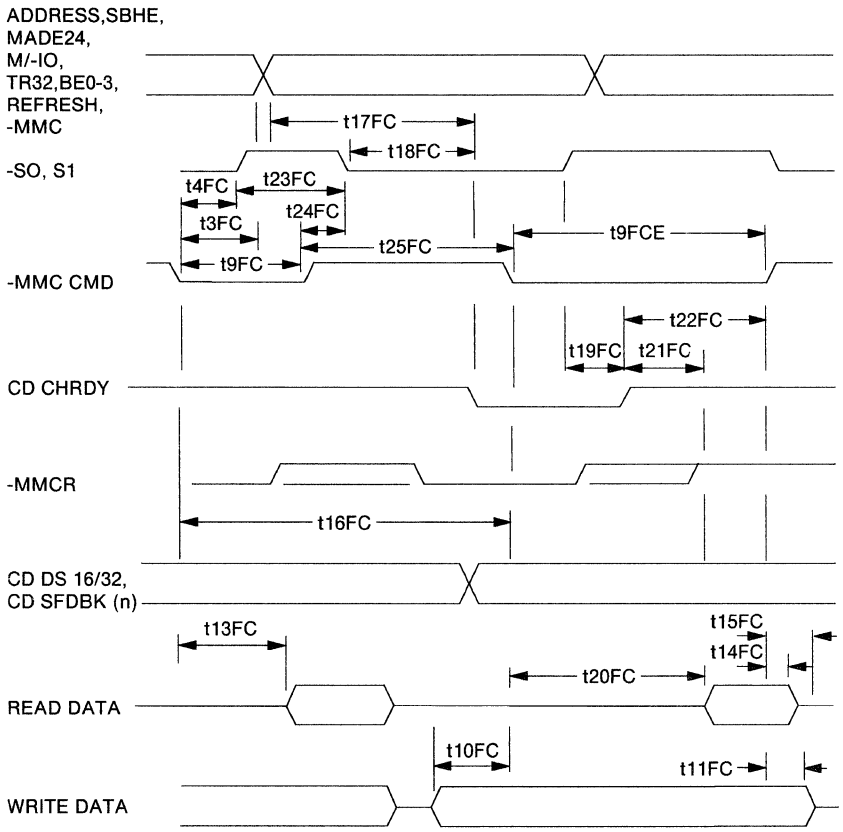
Figure 46. Matched Memory Cycle (No Waits)

Matched Memory cycles support 32-bit and 16-bit channel devices only. Eight-bit channel devices cannot run Matched Memory cycles.

Note: Adapters using this protocol can only be installed in the three 32-bit channel connectors and therefore sacrifice some *portability*. These adapters may not fit or operate in other systems.

When MMC is active, Matched Memory cycle 32-bit and 16-bit devices should not use -MADE 24, A0, A1, or -SBHE in logic that generates -MMCR, -CD DS 16/32, -SEL FBK, and -CD CHRDY.

Matched Memory Cycle Timing (One Wait)



Symbol	Description	Min/Max
t3 FC	ADDR hold from -MMC CMD active	20 / - ns
t4 FC	Status hold from -MMC CMD active	25 / - ns
t9 FC	-MMC CMD pulse width	85 / - ns
t9 FCE	-MMC CMD pulse width (Extended Cycle)	145 / - ns
t10 FC	Write data valid to -MMC CMD active	0 / - ns
t11 FC	Write data hold from -MMC CMD inactive	30 / - ns
t13 FC	Read data valid from -MMC CMD active	- / 60 ns
t14 FC	Read data hold from -MMC CMD inactive	0 / - ns
t15 FC	Read data off delay from -MMC CMD inactive	- / 40 ns
t16 FC	-MMC CMD active to next -MMC CMD active	180 / - ns
t17 FC	CD CHRDY valid from ADDR valid	- / 70 ns
t18 FC	CD CHRDY valid from Status active	- / 30 ns
t19 FC	CD CHRDY active from Status inactive	- / 9 ns
t20 FC	Read data valid from -MMC CMD active	- /125 ns
t21 FC	Read data valid from CD CHRDY active	- / 45 ns
t22 FC	-MMC CMD inactive from CD CHRDY active	65 / - ns
t23 FC	Status inactive pulse width	30 / -
t24 FC	-MMC CMD inactive to Status active	- / 5 ns
t25 FC	-MMC CMD inactive pulse width	85 / - ns

Figure 47. Matched Memory Cycle with One Wait

Matched Memory cycles support 32-bit and 16-bit channel devices only. Eight-bit channel devices cannot run Matched Memory cycles.

Note: Adapters using this protocol can only be installed in the three 32-bit channel connectors and therefore sacrifice some *portability*. These adapters may not fit or operate in other systems.

When MMC is active, Matched Memory cycle 32-bit and 16-bit devices should not use -MADE 24, A0, A1, or -SBHE in logic that generates -MMCR, -CD DS 16/32, -SEL FBK, and -CD CHRDY.

Adapter Design

This section provides some basic guidelines to design adapters for the Micro Channel Architecture 16- and 32-bit products. Topics include physical specifications, power requirements and limitations, and configuration program support.

The system board provides three types of connectors in the channel:

- 8- or 16-bit connector
- 8- or 16-bit connector with video extension
- 32-bit connector with matched memory extension (Model 80 only).

Plated connector contacts are not required for signals not used by an adapter. Three channel positions (1, 2, and 4) are provided on the Model 80 system board for 32-bit devices. Channel position 6 has the video extension connector. The video extension signal connector is used for display-only applications.

Physical Dimensions

The following figures show the dimensions of each type of adapter and the associated mounting hardware. The tolerances shown include all individual process tolerances and are noncumulative. The maximum height for components mounted on the adapter is 15 millimeters (0.6 inch) on the component (A) side. The maximum height for pins and components on the B side of the adapter is 2 millimeters (0.078 inches). Adapters using CMOS technology should be made with all plated connector contacts the same length to reduce the exposure of incorrect bias to modules.

General Design Considerations

Each designer is responsible for taking the necessary precautions to protect the safety of the end user, provide reliable operation of the device, and ensure the device does not interfere with the operation of the system or any other installed devices. This section is not intended to be an all-inclusive list of design considerations, but rather to point out some areas of design that may otherwise be overlooked.

Safety

Avoid exposed high voltage or current points, sharp edges, and exposed components that operate at high temperatures. Devices must not channel dc power outside of the system unit in any manner that violates Underwriter's Laboratory and Canadian Standards Association guidelines.

Note: Canadian Standards Association C22.2, paragraph 4.11.3, number 154 requires protection of conductors of external interconnecting cords and cables connected to secondary circuits.

IBM does not support installing or removing adapters or components when the system power is on.

Thermal

The system unit is cooled internally by low-volume forced air. Adapter designs must allow for adequate airspace between the adapters. Internal cables should be avoided as a mechanism for signal communication inside the system unit. These cables can interfere with the air flow. If internal cables are required, they must be positioned to minimize the impact on airflow. The maximum height for components mounted on the adapter should not exceed the dimensions specified under "Physical Dimensions." The adapter design should avoid clustering of high temperature components. No component should exceed its maximum thermal rating.

Electro-Magnetic Compatibility

Adhere to the following guidelines to reduce electro-magnetic compatibility (EMC) problems:

- The adapter end brackets make a continuous 360° connection to the outside "skin" of the system unit cabinet. A similar 360° connection to the inside skin should also be provided. The adapter bracket must not be used as a dc voltage return path, a logic-ground connection, or an audio ground connection.
- The adapter end bracket at the rear of the adapter is isolated from dc ground on the adapter. The bracket must be grounded through a screw connection to the system unit and designed as shown in Figure 61 on page 63.
- All connector ground pins must be connected to the interplane ground at the channel connector. Likewise, the +5 Vdc power must be immediately connected to the +5 Vdc power plane.

- All adapters must provide nonsegmented internal power and ground planes.
- Each surface-mount technology (SMT) module position should provide a decoupling capacitor pad with minimal connection inductance. Pin-in-hole (PIH) modules should be decoupled if they drive or contain edge-triggered logic. Capacitors can range between 0.01 and 0.10 μF and should be of a low inductance ceramic or layered design.
- Internal cables should be avoided as a mechanism for signal communication inside the system unit. The channel should not be extended outside of the system unit, except by an adapter.
- Clocks should be properly imbedded and terminated. When clocks, strobes, and handshakes are generated or received, care should be taken to control the rise-and-fall times to minimize radiation.
- External cables should connect through 360° shielded D-shell or equivalent connectors. Avoid the use of "pigtail" shield connections. Shield terminations should be connected to the external shield of the cable connector. Do not bring the shield through the connector and connect it to either logic ground or the inside skin of the cabinet.
- High-current power within the system unit should provide adjacent or sandwiched return paths to allow the maximum cancelation of radiated magnetic fields by the mutual coupling between the supply and return lines.

Diagnostics

All writable registers typically are readable at the same address. External interfaces typically include 100 percent diagnostic wrap capability by electronic switching or an external wrap tool.

Design Guidelines

Adapters designed to be used in the Micro Channel architecture must comply with the following design guidelines:

- Each I/O adapter design must decode all 16 bits of the I/O address.
- Each memory adapter design must decode all 24 bits of the memory address and MADE 24.
- Each 32-bit memory adapter must decode all 32 bits of the memory address if MADE 24 is inactive.
- Each adapter design must replace the function of switches and jumpers with registers and incorporate POS logic.
- Each adapter must output a card ID to the data bus when interrogated.

  To minimize the number of drivers required, only the logical 0 bits in the card ID need to be driven. This provides 39,202 combinations with 8 drivers or less.

  The following figure shows the recommended ID values for vendors.

ID	Definition
0000	Device Not Ready
0001 to 0FFF	Bus Master
5000 to 5FFF	Direct Memory Access Devices
6000 to 6FFF	Direct Program Control (Including Memory-Mapped I/O)
7000 to 7FFF	Storage*
8000 to 80FF	Video
FFFF	Device Not Attached

* Multiple-function adapters containing storage typically respond as storage.

Figure 48. ID Assignments

- Each enabled adapter must return a 'card selected feedback' signal (CD SFDBK) to the system microprocessor when an access is made to the address space of the adapter, or when the adapter is selected by arbitration level. CD SFDBK must not be generated when the '-card setup' (-CD SETUP) line is active.
- Each adapter design must be capable of de-gating all outputs to the system board (including CD SFDBK, -CD DS 16/32, interrupts, and so on) if the least-significant bit of the option select word at address hex 0102 is set to 0.
- Each adapter design must implement an open collector driver (or tri-state driver gated negative active) to drive the interrupt request line. The design must also implement a status register (readable at an I/O address bit position) that remains active when the interrupt is set and stays active until reset by the service routine. The adapter must hold the level-sensitive interrupt active until it is reset as a direct result of servicing the interrupt. The service routine must not reset the interrupt controller until after the interrupt bus signal has been reset by the adapter.
- Following a reset, each adapter must set bit 0 (Card Enable) of its POS register 2 to 0.

- If applicable, the adapter can have the ability to reside at an alternate address (corresponding to one selected by switches on a Personal Computer-type adapter).
- All adapters must provide system configuration (.adf) files on a 3.5-inch diskette.

Feature Card Design Considerations

The Personal System/2 feature cards are dramatically different from the feature cards used in the IBM Personal Computers. The card height and length has been reduced. The bus connector to the system has been completely redesigned using 0.050 inch connector technology.

Several reasons contributing to the changes are increased performance, advanced technology, FCC Class B environment and reduced system profile. The feature cards for IBM Personal Computers and the IBM Personal System/2 are physically and electrically incompatible with each other. It is hoped that the information that follows will prevent problems and delays in the development of feature cards to be installed into the new Personal System/2.

Electro-Magnetic Compatibility (EMC)

The Personal System/2 system units have been certified to comply with the limits for a FCC Class B computing device, pursuant to Subpart J of Part 15 of the FCC rules. Only peripherals (computer input/output devices, terminals, printers, etc) certified to comply with the Class B limits may be attached to this computer when this computer is operated in a residential environment. Operation with non-certified peripherals is likely to result in interference to radio and TV reception.

Card Form Factor

The raw card physical size is 3.475 X 11.5 inches.

The physical size of the feature cards for the Personal System/2 is influenced primarily by the physical size of the system. The secondary reason for the exact dimension is the number of cards that can be obtained from a standard size panel.

The 3.475 inch height of the card is critical. If the height of the card is increased, the card will interfere with the covers. If the height is decreased, the circulation of air to cool the cards for the system could be reduced.

The length of the card is not as critical but caution is required.

Example 1:
The normal raw card is 11.5 inches in length and the retainer that fastens to the end of the card extends the overall length to approximately 12.2 inches. The retainer sits in a channel that will support the card and prevent the card from tilting sideways, coming into contact with adjacent cards.

Example 2:
Discussions have taken place on the subject of a half size card. Although we have not built one of these cards, we feel the overall length should still be approximately 12.2 inches. If a raw card was 6 inches in length, a retainer that would extend the overall length to 12.2 inches would be required. The main reason is that air flow through the feature cards could be altered if a short card was installed.

Example 3:
Likewise, the discussion about a 12 inch raw card took place. IBM does not support a card that is greater than 11.5 inches in length.

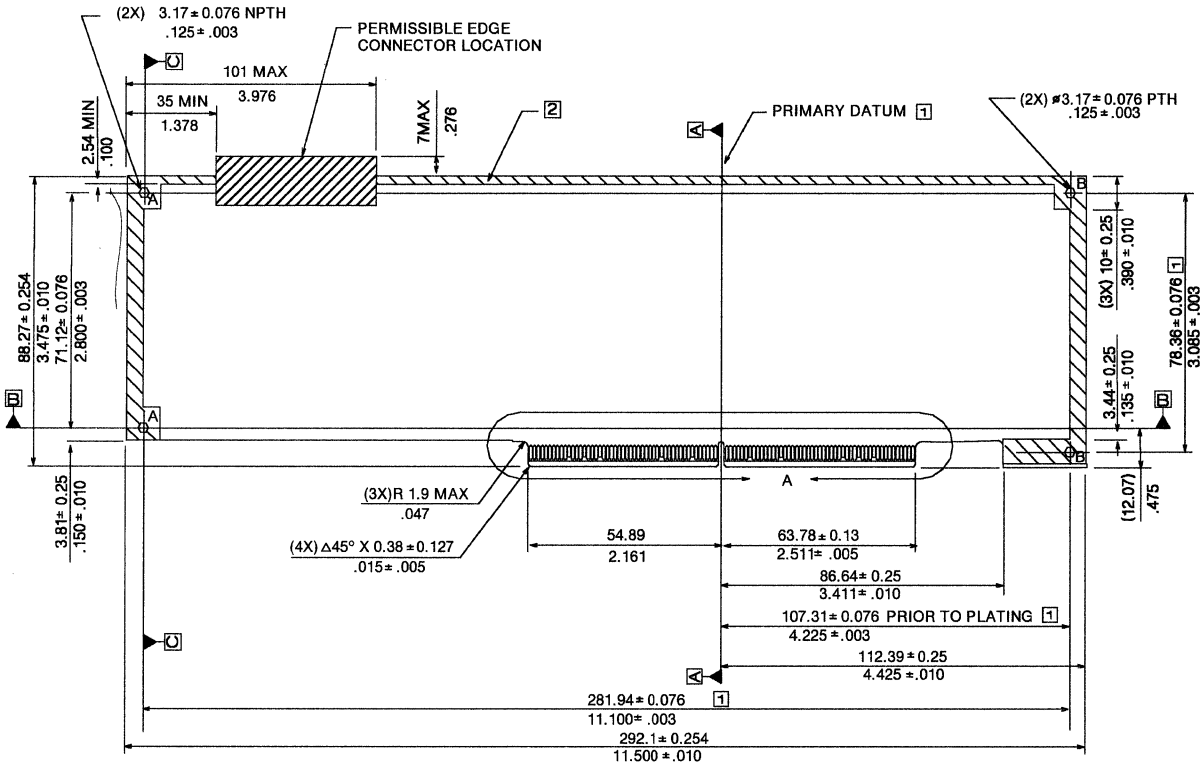


Figure 49. Personal System/2 32-bit Adapter Card

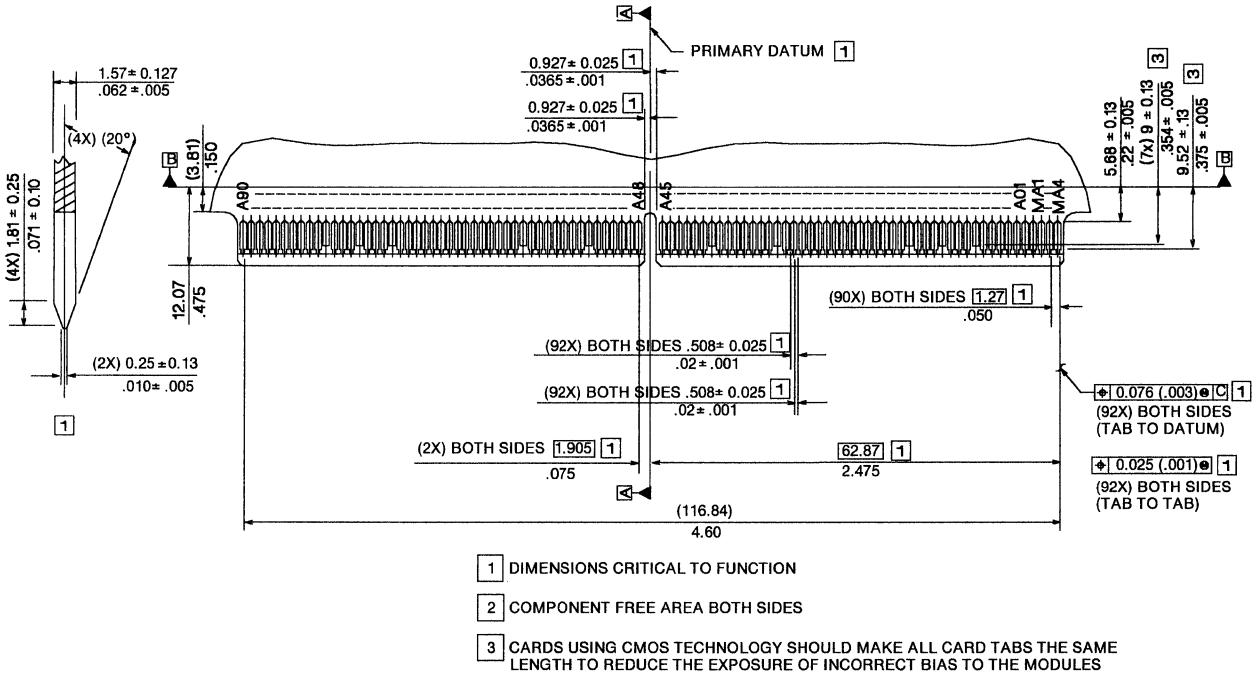


Figure 50. Tab Area for 32-bit Interface

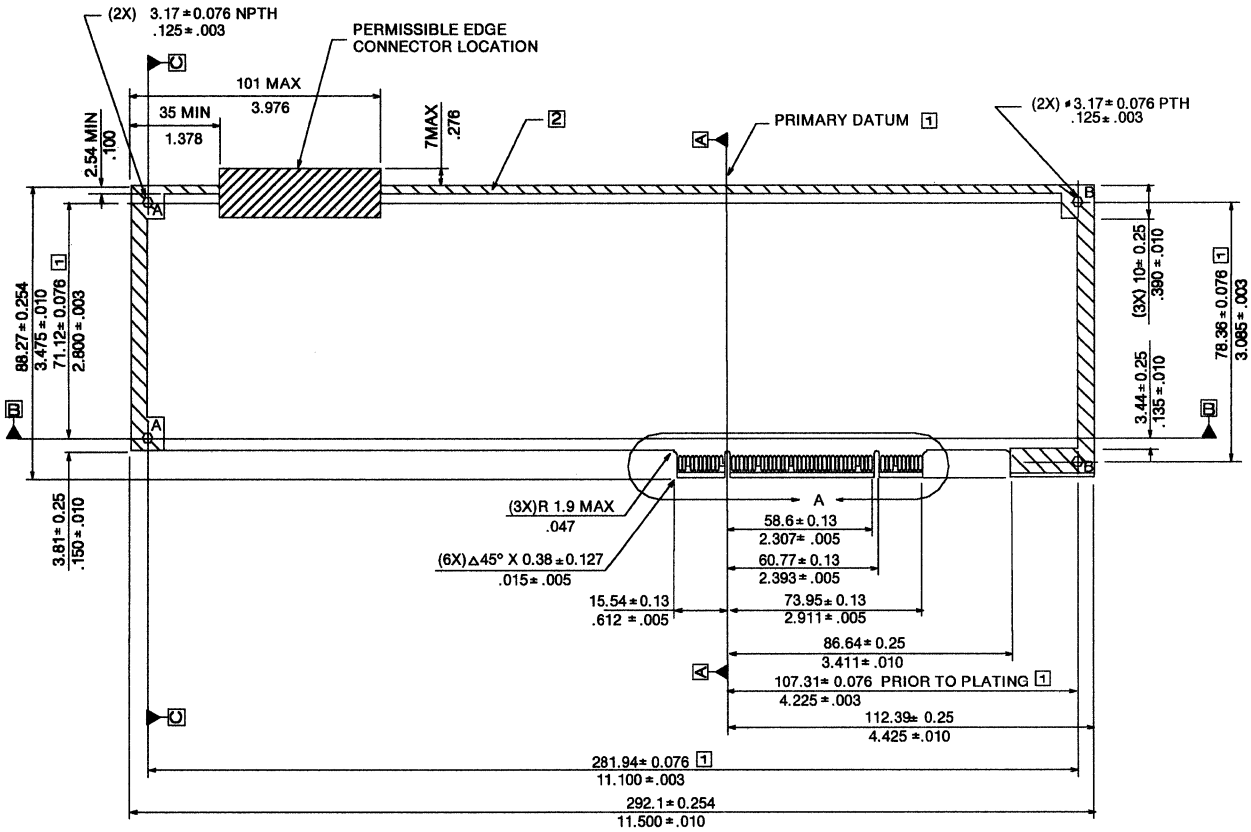


Figure 51. Personal System/2 Extended Video Card

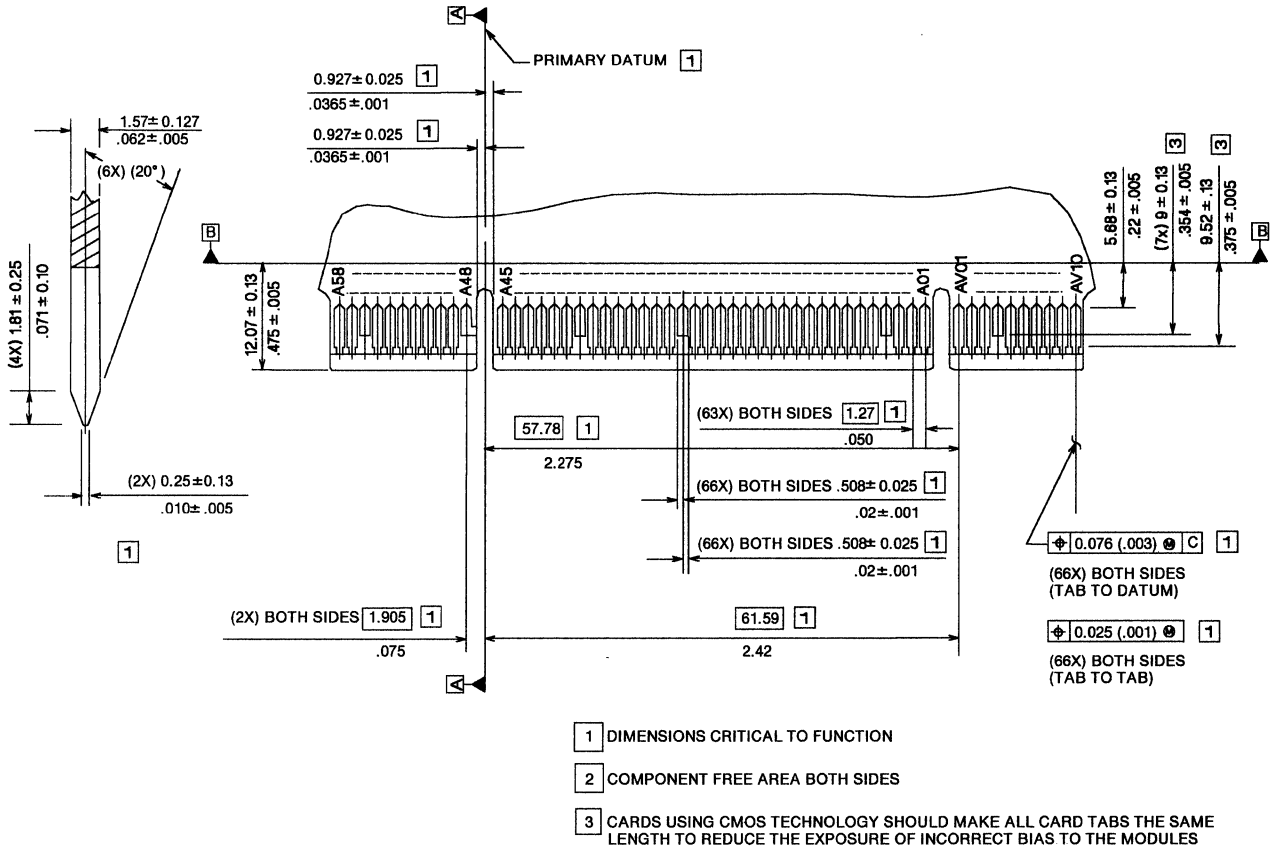


Figure 52. Tab Area for Extended video

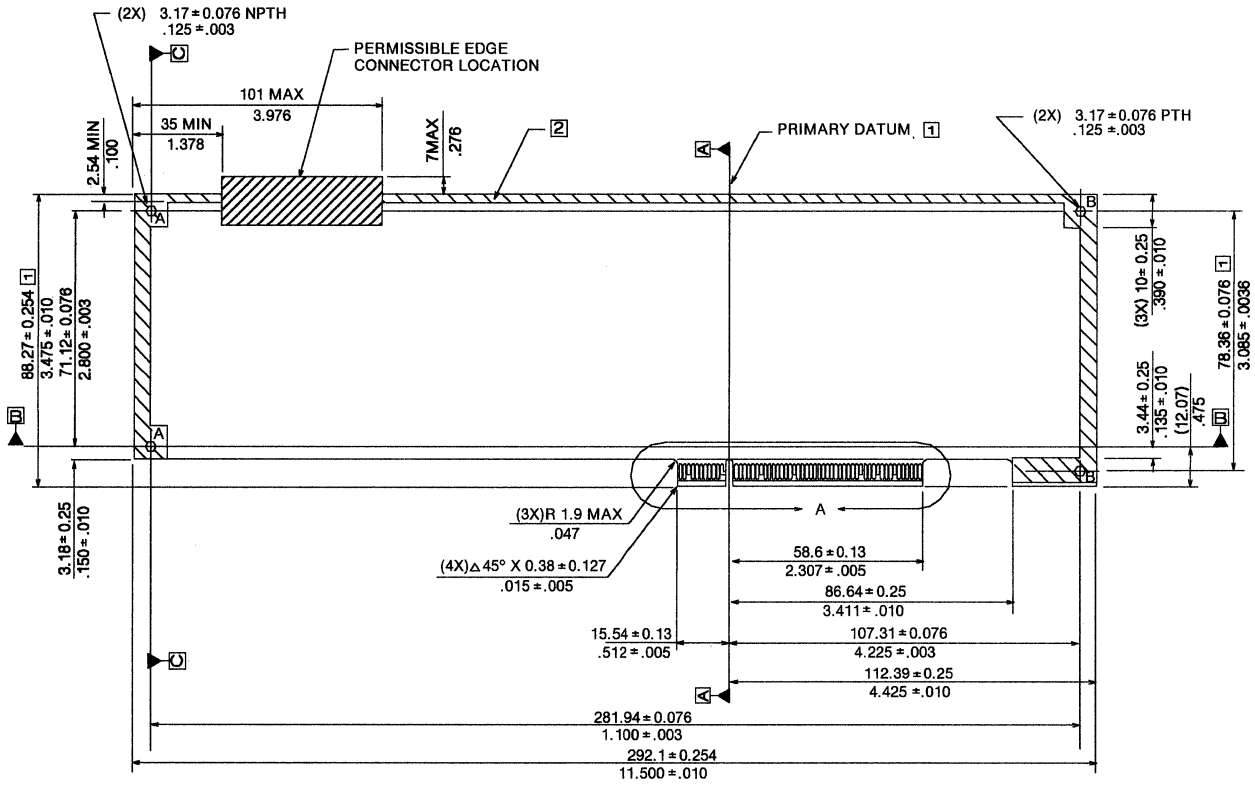


Figure 53. Personal System/2 Standard Feature Card

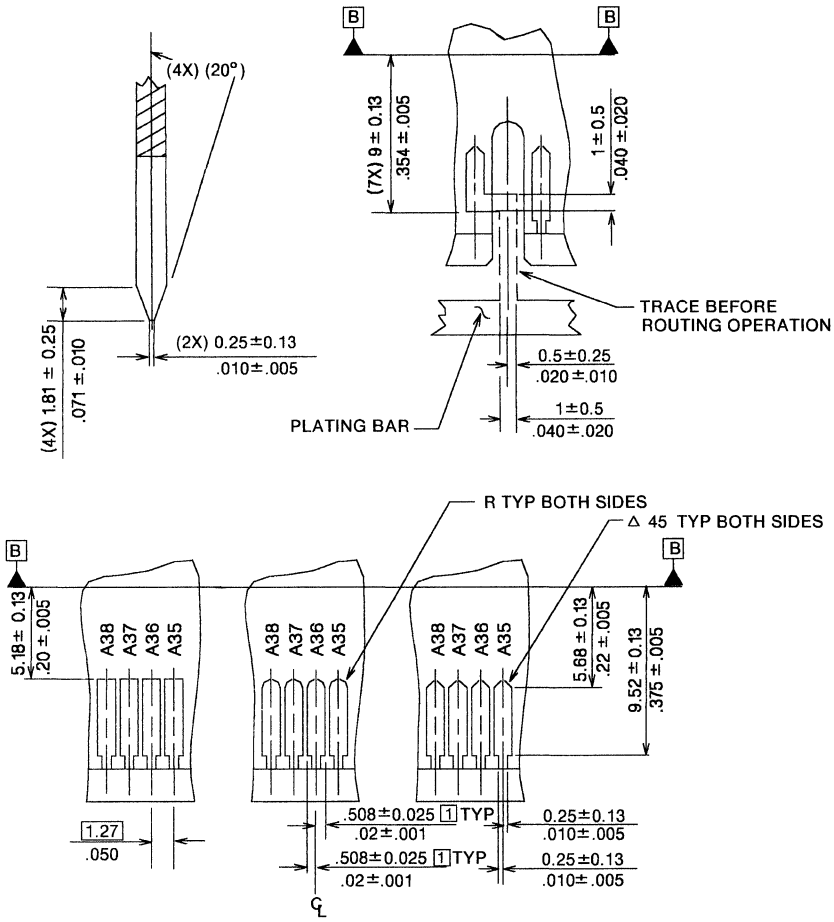


Figure 54. Connector (Common Detail)

Raw Card Thickness

The bus connector is designed for a feature card thickness of 0.062 inch +/- 0.005 inch. If a card were built with a thickness less than specified, the card would not have the necessary wiping action on the pins to make good contact.

Assembled Card Thickness

The distance from card to card on the system bus is 0.850 inch. In addition to raw card, the maximum component height is 0.600 inch and the maximum length of pins or components on the reverse side is 0.080 inch. The maximum total thickness of the assembled card is 0.742 inch. If a daughter card concept is used, the maximum total thickness is also 0.742 inch. The remaining space is required to circulate air through the cards.

Primary Datum

The primary datum is located in the center of the slot in the tab pin area. This is a critical location and should not be moved. During the development of the feature cards for the Personal System/2, several discussions took place to move the datum to the edge of the card. Granted, moving the datum to the edge of the card would ease the job of manufacturing the raw card. However, the ability to hold dimensions and tolerances on the most critical part of the card, the slot and tab pins, would be extremely difficult. The dimension of the slot is critical because of the 0.050 inch bus connector. This slot is required on all feature cards that plug into the Personal System/2. See the previous figures for the dimensions of the datum.

Examples

Tolerance For the Slot: The tolerance on EITHER side of the primary datum is $\pm$0.001 inch. You cannot have 0.0 inch tolerance on one side of the primary datum and +0.002 inch tolerance on the other side.

Tolerance For The Tabs: The tolerance from the primary datum to the FIRST tab on either side of the datum is $\pm$0.003 inch, on both sides of the card.

Example: The first tab that is located next to the slot is also the tab that is closest to the primary datum. The second tab is 50 mils from the tab that is adjacent to the slot and therefore the location is 50 mils $\pm$0.5 mils from the first tab.

The third tab is 100 mils from the tab that is adjacent to the slot and therefore the location is 100 mils $\pm$0.5 mils from the first tab, and so forth.

Tab Pins - Necked Down

The tab pins on the feature card are on 0.050-inch centers. The width of the tab pins is 0.040-inch. That leaves 0.010-inch clearance between the tabs. During the beveling operation on the bottom of the tab area, the gold will have a tendency to "smear" into the adjacent tab pin. The method we use is to neck down the width of the tab pins from 0.040 inch to 0.020 inch just above the area where the beveling operation would take place. This will significantly reduce the possibility of shorting the tabs during the bevel operation. Refer to Figure 54 on page 55 for results of the neck-down operation.

Beveled Edge

The feature cards are inserted into a connector on the system board that will have 112 to 182 connector pins. Inserting a card with that many pins became a concern. Several cards were built and tested with 45, 30 and 20 degree bevels. Because insertion force was a prime consideration, the 20 degree bevel was selected. See Figure 54 on page 55 for the detail of the beveled edge.

Internal Power Planes

Internal power planes help reduce EMC problems and should be planned from the initial stages of card development. The power planes can be either solid or grid format. All feature cards must provide nonsegmented internal power and ground.

During raw card layout, be careful with clearance holes on the power planes. A minimum of 0.006 inch of copper should be located between clearance holes that are adjacent to each other. What could happen is that several vias could be located in a straight line segmenting the power plane, which could result in a power distribution and/or EMC problem.

Example

Vias are located above the tab pin area. The vias are 0.018 inch in diameter, on 0.050-inch centers, across most of the tab area. The clearance holes on the power planes are 0.054 inch in diameter. The problem is that the voltage and ground entering the card and the card logic that needs the power are segmented from each other except for the area around the corners of the tab area. If a clearance of 0.044 inch in diameter is used, then 0.006 inch of copper would be located between the vias and not segment the power plane.

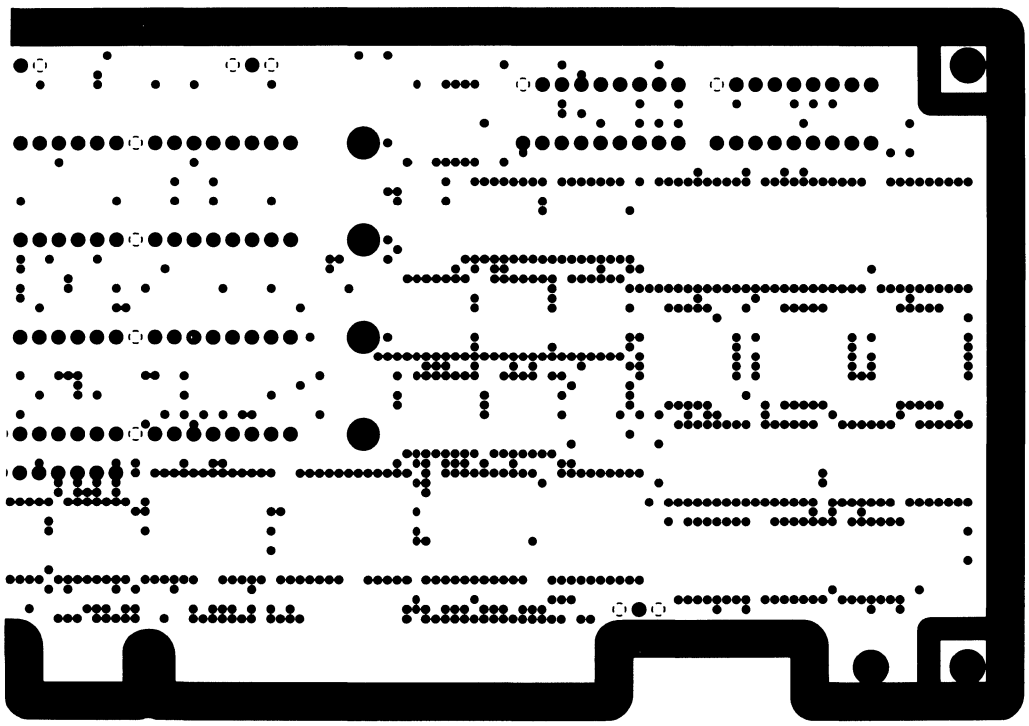


Figure 55. Power Plane Clearance holes

Internal Power Planes Into Tab Area

As mentioned earlier, EMC is extremely important in the design of the Personal System/2. Extending the internal power planes into the tab pin area helps to reduce the possibility of reflections.

CAUTION:
If the power plane is extended to within 0.100 inch of the edge of the card prior to the beveling operation, the power planes will be exposed after the 20 degree bevel operation. If someone accidentally installs the card with system power on, the exposed planes could short all the tabs to voltage or ground. The recommendation is to extend the power planes into the tab area approximately 0.150 inch, which is near the vertical center of the tab pins.

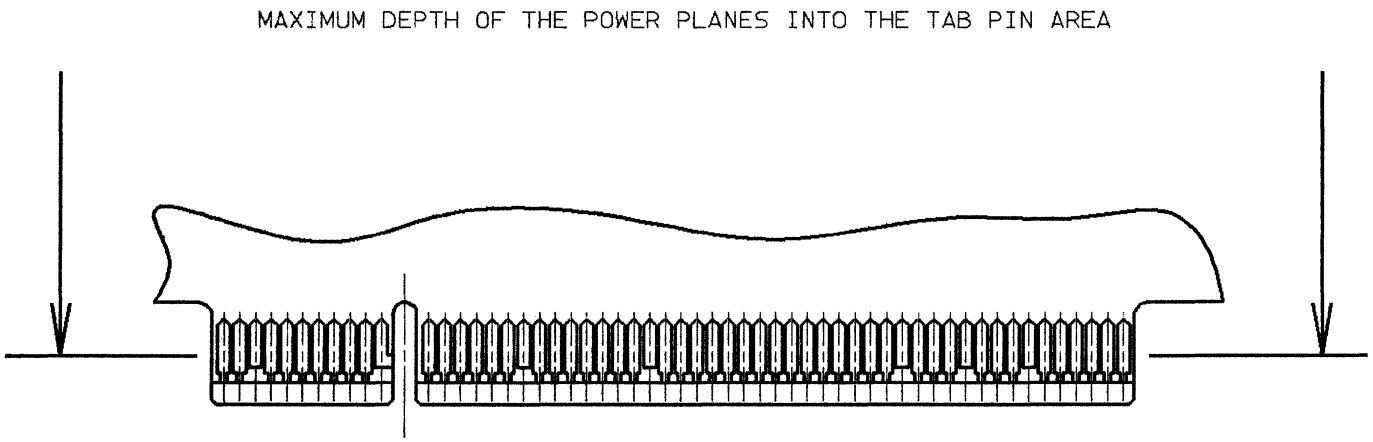


Figure 56. Power Plane Extended into Tab Pin Area

Short Tabs on 5-Volt Pins

On most of the feature cards designed for the IBM Personal System/2, the 5-volt pins are shorter than the remaining pins. The intent is to reduce the possibility of damaging feature cards when the cards are accidentally installed while the system power is on. Depending on technology, new feature cards may not take advantage of the short voltage pins and therefore, have all tab pins the same length.

The plating process for the short tab pins is relatively simple.

Pin A48 is a 5-volt pin located adjacent to the slot (refer to Figure 54 on page 55). Copper extends to the side of pin A48 into the area where the slot will be located and down to the commoning bar for the plating operation. Pin A48 is also connected to the 5-volt power plane with a wide land. All of the 5-volt pins should use a wide land when possible.

During the plating process, the current flows from the commoning bar to pin A48, to the 5-volt power plane, to the remaining 5-volt tabs. When the card slot is created, the land that was connected to the commoning bar will be removed. There is a very small area of copper on the side of pin A48 that is exposed, but when the card is installed in the system, the area is protected.

Card Frame Ground/DC Ground Isolation

EMC is the concern of all feature cards, but the feature card that communicates outside the system environment will have the greatest concern with EMC. In the Personal System/2, every attempt has been made to separate frame ground noise from the DC ground on the logic cards. The frame ground and DC grounds are connected at a common point in the system to reduce the EMC concerns. Therefore, two situations need to be discussed separately.

Cards Without External Interface

The feature cards that are strictly internal to the system (for example, the memory card) will have only the metal bracket on the I/O end of the card. Because the bracket is physically in contact with the frame of the system, the bracket must be separated from DC ground. The holes located in the top and bottom corners of the card are plated-through holes. The voltage and ground planes are physically separated by a 0.120 inch break in the power planes in both corners and therefore isolating frame noise and DC grounds on the card. In Figure 57, areas in black designate the areas to be free of copper.

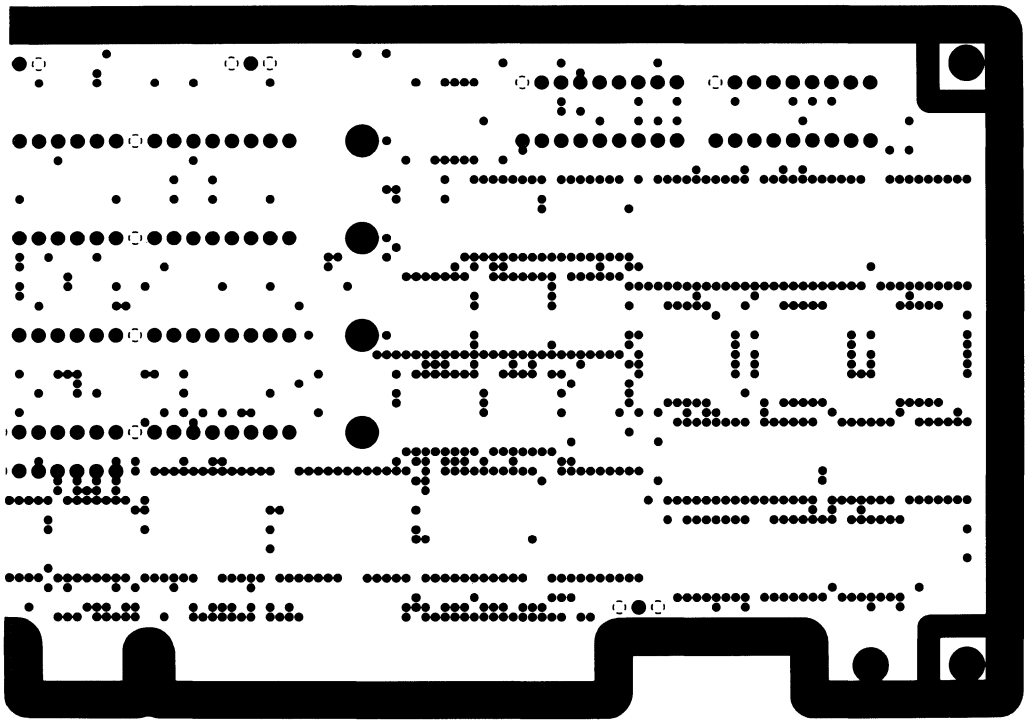


Figure 57. Power Isolation Without External Interface

Cards With External Interface

The feature cards that interface to external devices outside the system (for example, the Asynchronous, Multi-Protocol, and Modem) will have the metal bracket, external connectors and cables to be concerned with EMC. The bracket is physically in contact with the frame of the system and the external connectors, therefore the bracket and the connector hardware must be separated from DC ground.

Again, the holes located in the top and bottom corners of the card are plated-though holes. The voltage and ground planes are physically separated by a 0.120-inch break in the power planes, which includes both corners and the connector mounting hardware that would secure the external connectors. The easiest way to isolate the grounds is to have a break that runs from the top to the bottom of the card in the power planes. This method will insure the isolation of frame ground and cable noise from DC ground. In Figure 58, areas in black designate the areas to be free of copper.

From a safety standpoint, any static electricity or minimal lightning disturbance that may come in on the cable shielding would be connected to system ground.

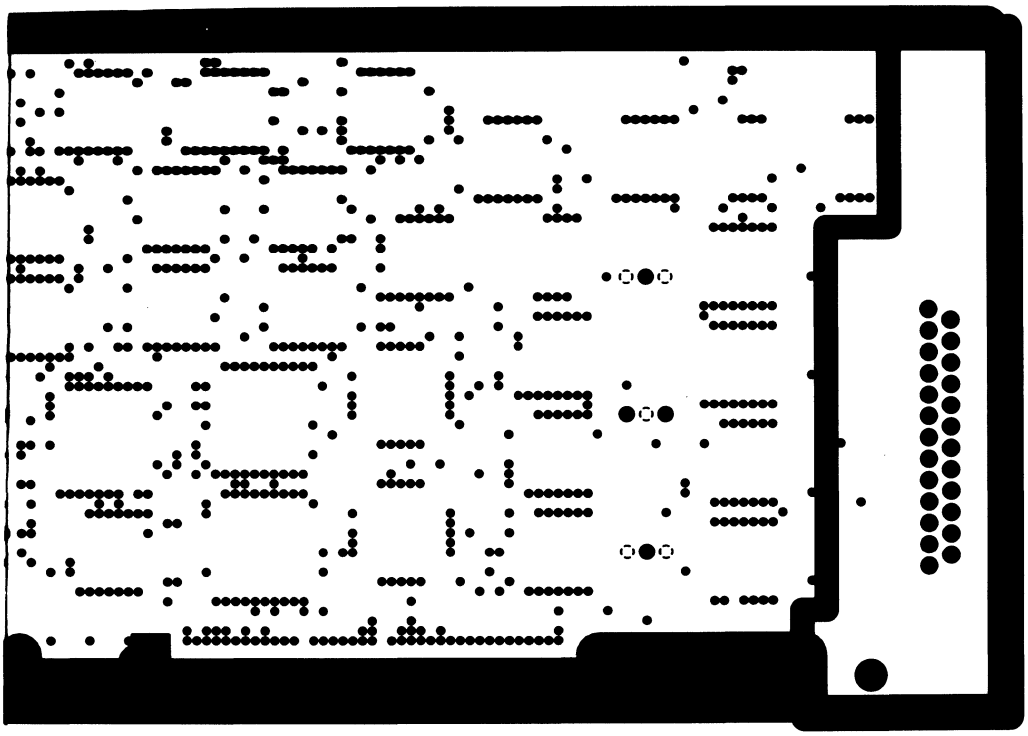


Figure 58. Power Isolation With External Interface

Analog Ground Isolation

Cards with analog circuitry must use caution to insure the analog ground is separated from the digital ground on the card. The grounding configuration for the analog circuitry will depend on the amount of grounding required for the card.

The Personal System/2 standard feature card has approximately 16 ground pins on the Micro Channel. The System Technical Reference Manual has a complete list of the ground pins. The "Audio Ground" pin on the channel should not be used as an analog ground.

The analog engineer for the card will have to determine how many of the ground pins on the Micro Channel will be required for the analog circuitry. The analog and digital ground for the card will be common at the system board level.

The large white area on the right side of the card is analog circuitry (refer to Figure 59). In this particular situation, the EMS characteristics were reduced by removing the power planes in the area of the analog circuitry. This is a decision that you will have to make, if you are planning to use analog.

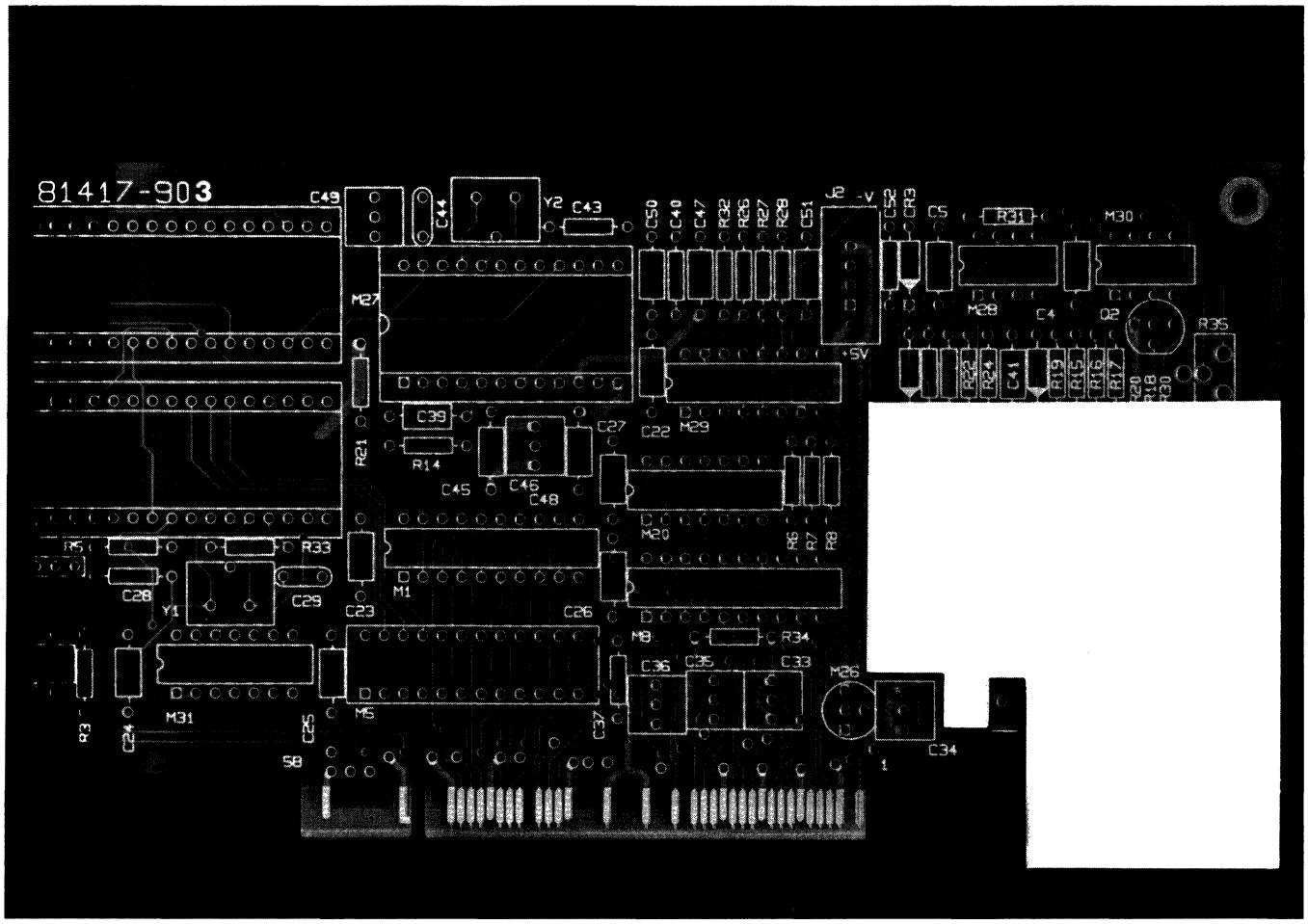


Figure 59. Analog Power Isolation

Card Layout With Metal Bracket

The metal bracket covers 0.200 inch of the card when installed. When laying out the card, do not install any components near the I/O end of the card that would interfere with the installation of the bracket. During the wiring operation, be sure not to route lands under the bracket because the bracket is located just above the circuit board.

If the bracket was slightly bent, the bracket could short to the lands. In addition, be careful when placing components near the bracket whose leads could short to the bracket. See Figure 60 and Figure 61 on page 63.

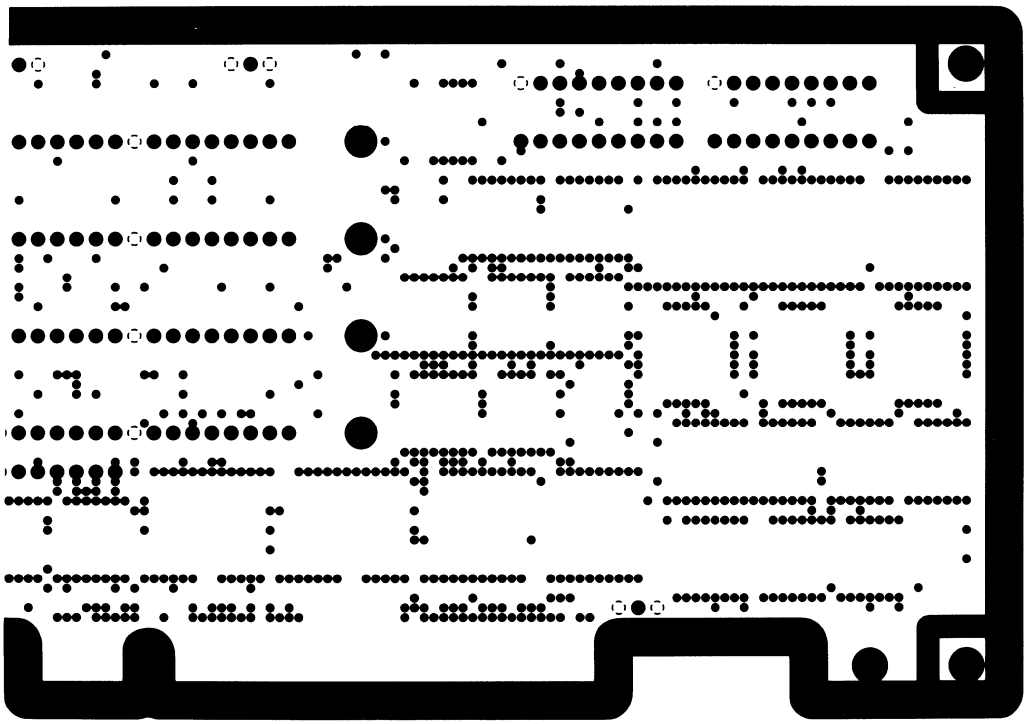
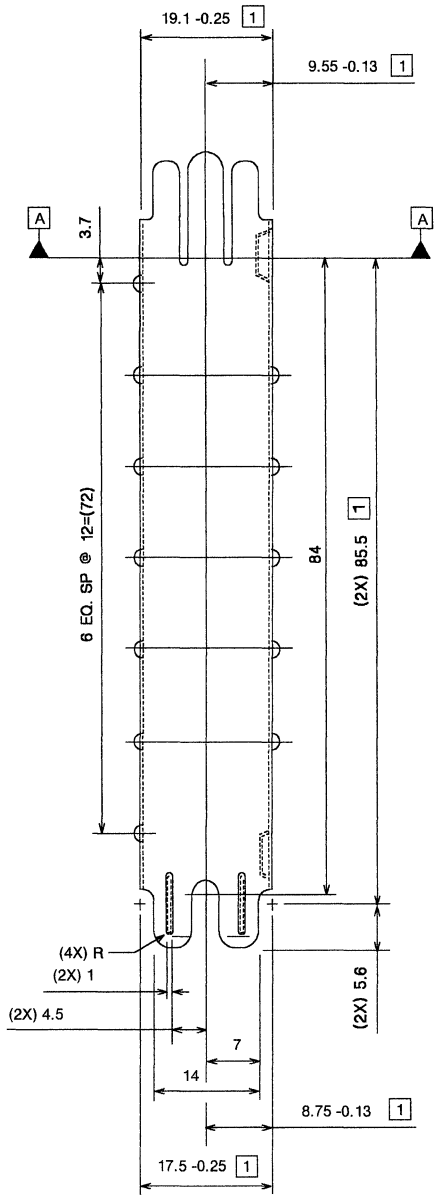


Figure 60. Card Layout for Metal Bracket



1 DIMENSION IS CRITICAL TO FUNCTION.

Figure 61. Metal Brackets (1 of 4)

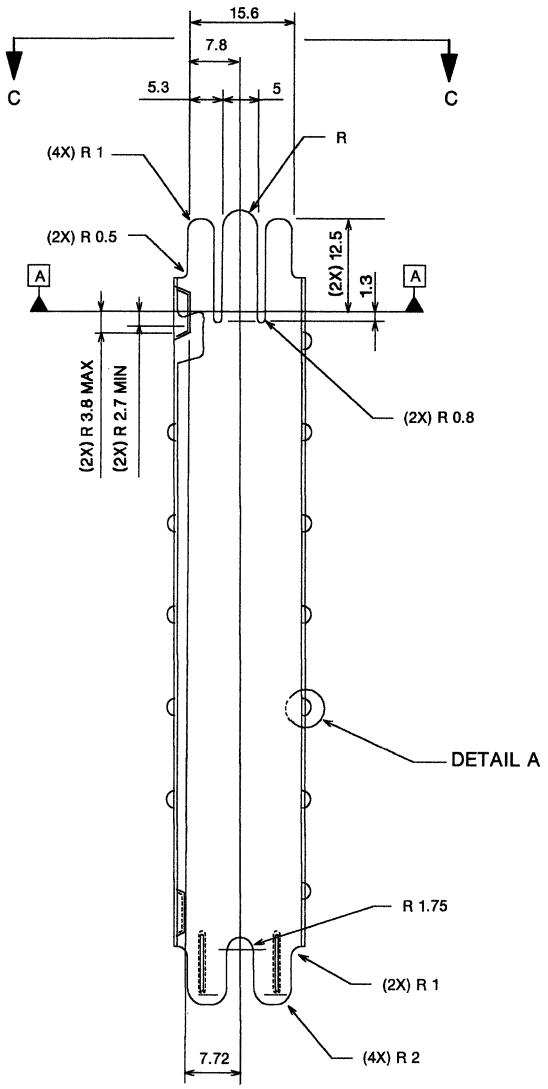


Figure 62. Metal Brackets (2 of 4)

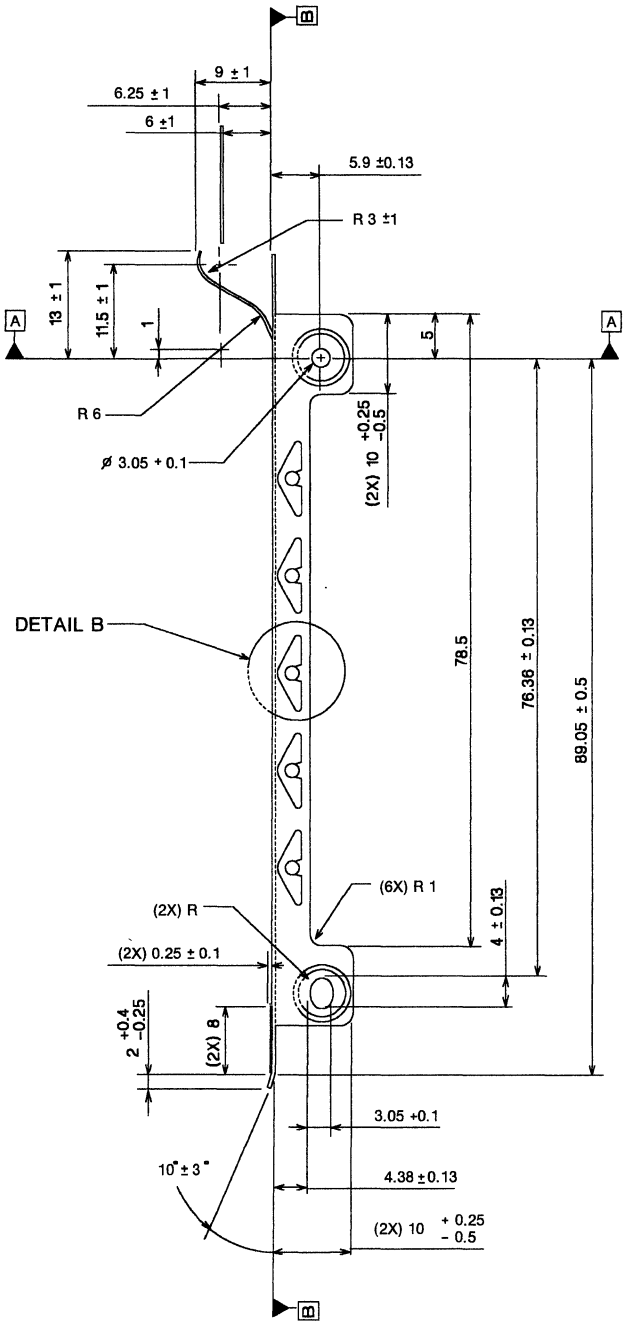


Figure 63. Metal Brackets (3 of 4)

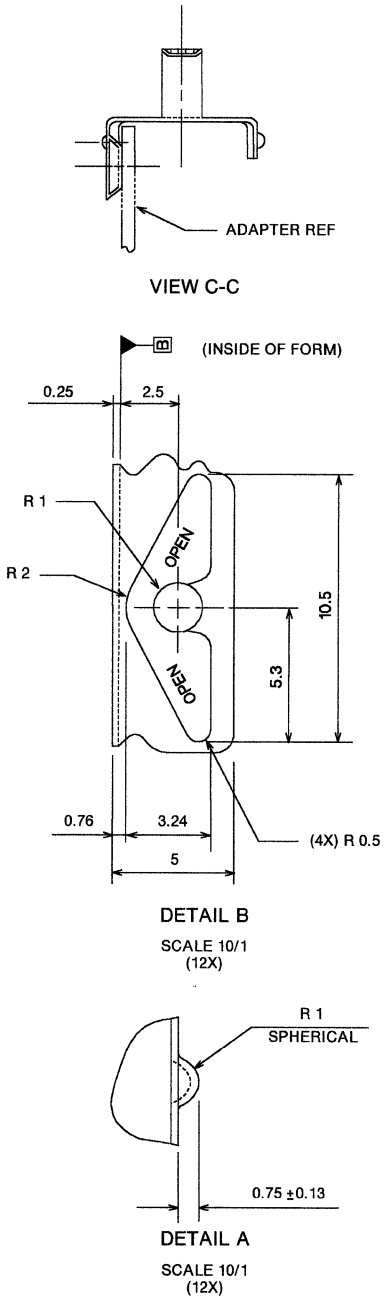


Figure 64. Metal Brackets (4 of 4)

Card Retainer

The opposite end of the card from the metal bracket is where the plastic retainer is located. The holes for the retainer are located in the upper and lower corners of the card. The holes for the retainer are non-plated through holes (NPTH). See Figure 65 and Figure 66 on page 68.

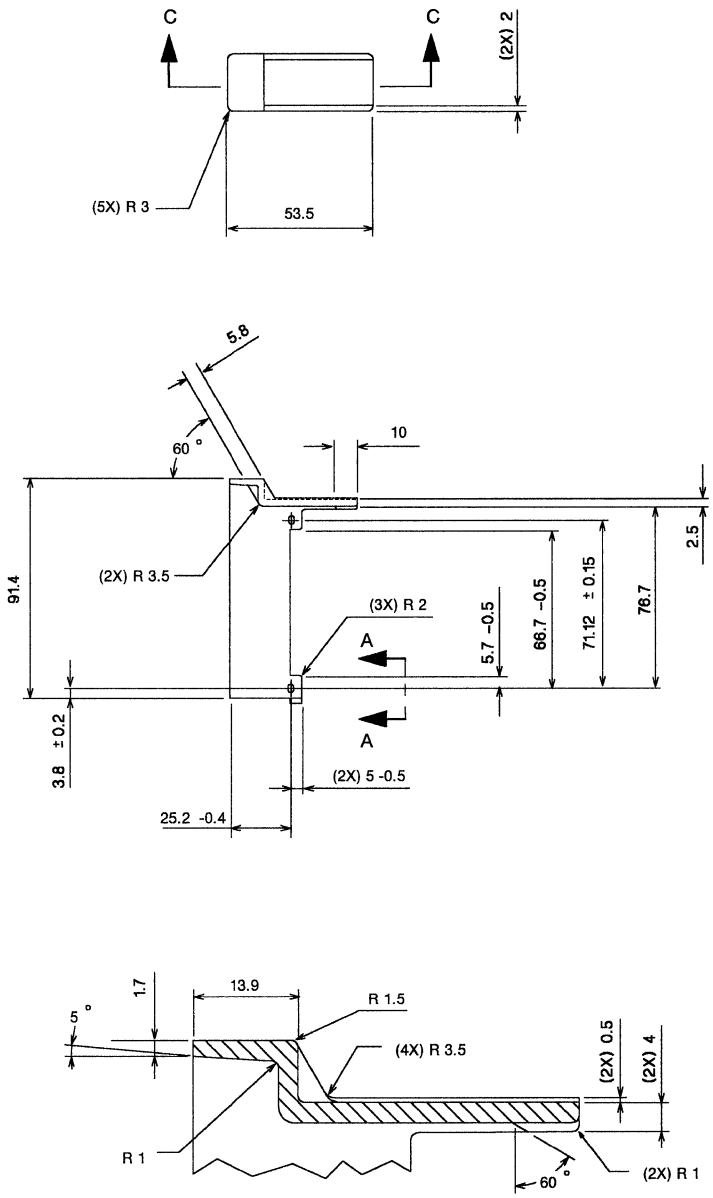


Figure 65. Card Retainers (1 of 2)

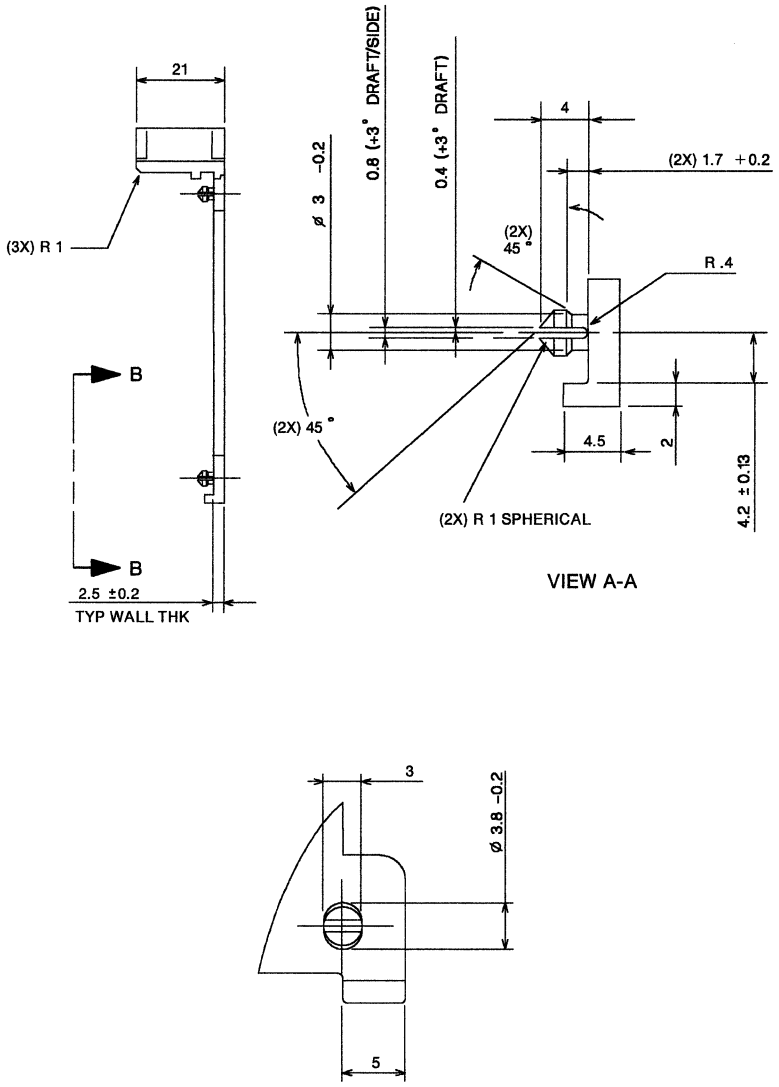


Figure 66. Card Retainers (2 of 2)

Top Card Connectors

Cards that require a top card connector must use caution. The raw card drawings indicate an area that is used for this purpose. The top of the connector must not extend beyond 0.276 inch above the top edge of the card. See Figure 49 on page 50. Above this point the connector will interfere with the baffle and/or the covers. If the connector is located at any point other than that which is indicated, the cabling will impact airflow through the system.

External Connector Shielding

All external cables should connect through a 360 degree shielded connector. The shielding must be connected to the AC ground which could be the metal bracket or the ground plane that has been segmented from the DC ground.

Figure 67 shows four D-shell connectors of different sizes. The 360 degree shielding is comprised of the area of the card mounted connector which is flush with the metal bracket.

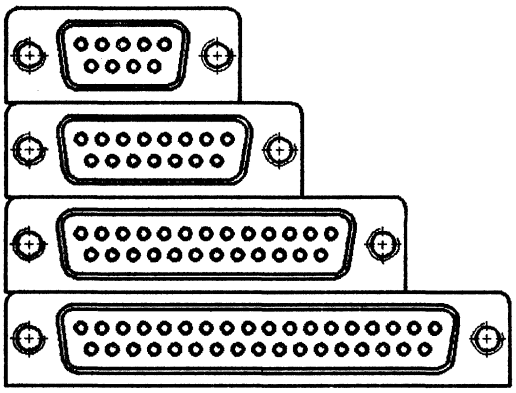


Figure 67. D-Shell Connectors

Figure 68 shows a single connector and the shielding has two tabs that fold over the top edge of the connector base to secure the shield. The edge that protects the pins is also a portion of the shielding and will make physical contact with the metal bracket.

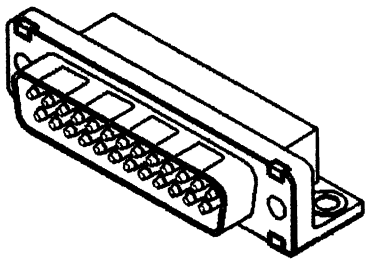


Figure 68. Shielded D-Shell Connector

Figure 69 shows the metal bracket. Look at the opening where the D-shell would be inserted, the tabs on the side are the contact points that physically come in contact with D-shell connector. The holes located above and below the opening are where the standoffs will secure the connector to the bracket.

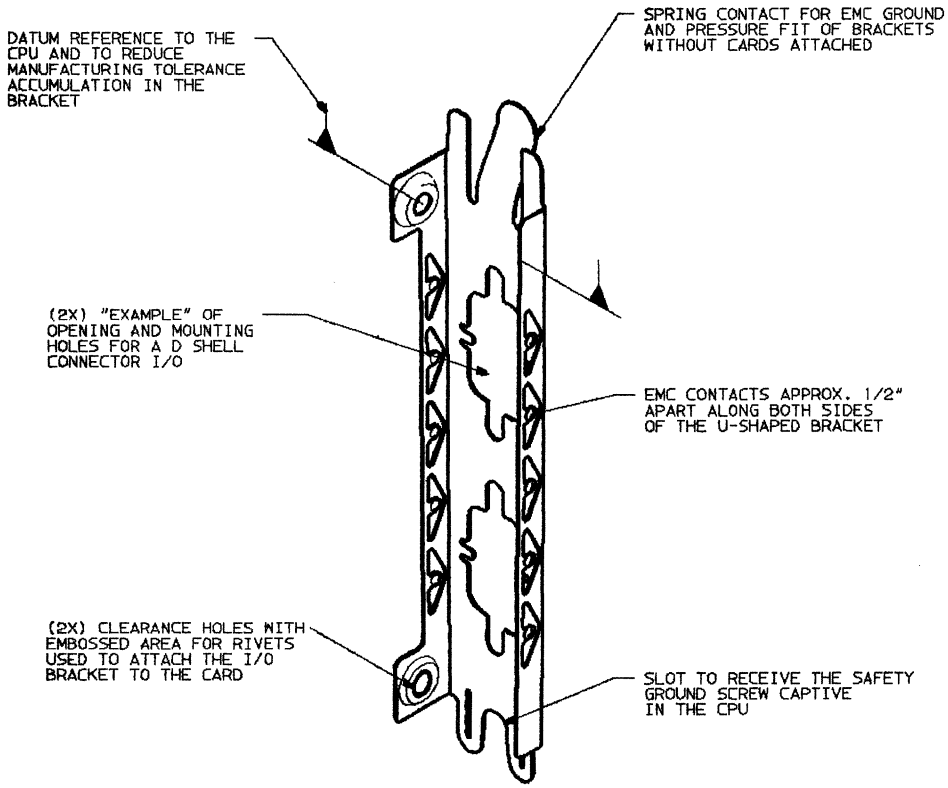

DATUM REFERENCE TO THE CPU AND TO REDUCE MANUFACTURING TOLERANCE ACCUMULATION IN THE BRACKET
SPRING CONTACT FOR EMC GROUND AND PRESSURE FIT OF BRACKETS WITHOUT CARDS ATTACHED
(2X) "EXAMPLE" OF OPENING AND MOUNTING HOLES FOR A D SHELL CONNECTOR I/O
EMC CONTACTS APPROX. 1/2" APART ALONG BOTH SIDES OF THE U-SHAPED BRACKET
(2X) CLEARANCE HOLES WITH EMBOSSED AREA FOR RIVETS USED TO ATTACH THE I/O BRACKET TO THE CARD
SLOT TO RECEIVE THE SAFETY GROUND SCREW CAPTIVE IN THE CPU

Figure 69. Metal Bracket

Figure 70 shows a bottom view of the connector. The shielding is folded under the connector and back to the eyelet where the connector is secured to the card. The eyelet also insures the connector shield is tied to AC ground.

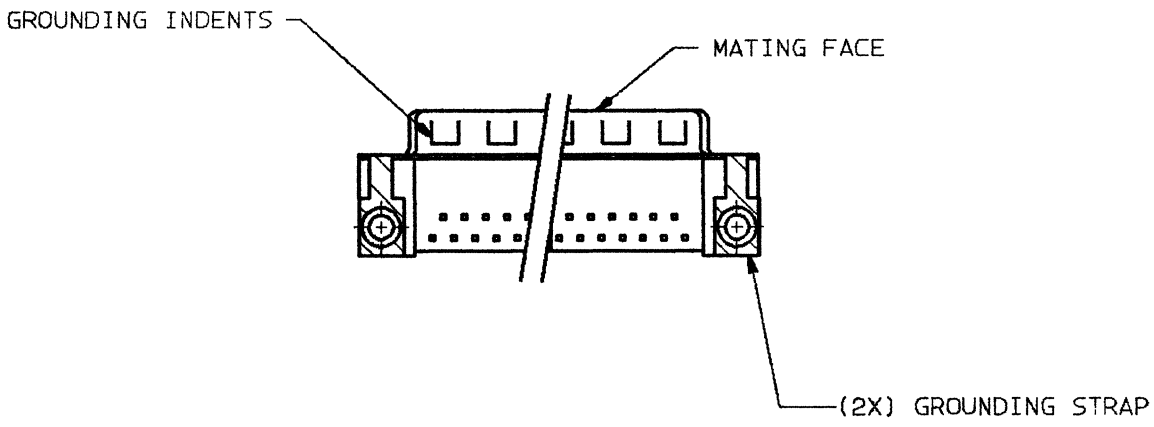


Figure 70. Grounding Strap Beneath Connector

Component Placement (Bus Interface) Micro Channel

The timings on the Personal System/2 interface are more critical than on IBM Personal Computers. The components that interface directly to the system bus must be as close to the bus as physically possible.

During component placement, these components should be manually placed to insure close proximity to the channel connector. Lands connected to the system bus are a prime concern and should be as short as possible, preferably less than two inches in length.

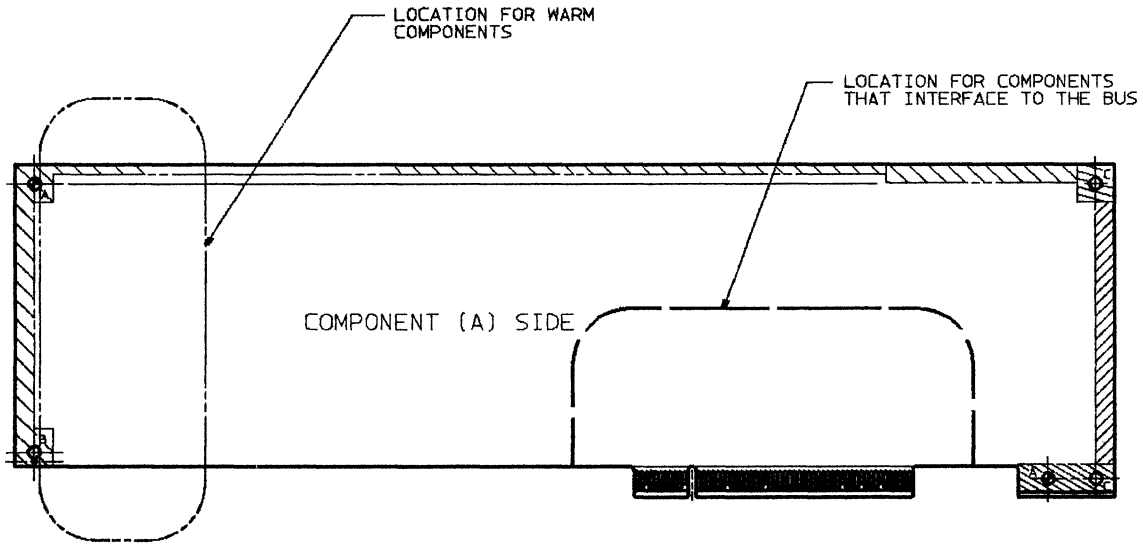


Figure 71. Component Locations for Thermal and Interface

Thermal Considerations

The system unit is cooled internally by low volume forced air. Allow for adequate airspace between the adapter cards. The adapter design should avoid clustering of high temperature components. No component should exceed its maximum thermal rating.

Components that may operate warm should be located near the front of the system (the end opposite the metal bracket). This area is near the opening in the front bezel that allows air to pass through the system. The components that operate warm should be manually placed at the beginning of the layout operation.

Power Availability Per I/O Slot

Supply Voltage	Maximum Statistical Current per 8/16 Bit Slot	Maximum Statistical Current per 32 Bit Slot
+ 5.0 Vdc	1.6 A	2.0 A
+12.0 Vdc	0.175 A	0.175 A
-12.0 Vdc	0.040 A	0.040 A

Maximum Statistical Current =

$$(I_{TC_1} + \cdots + I_{TC_n}) + \sqrt{((I_{MC_1} - I_{TC_1})^2 + \cdots + (I_{MC_n} + I_{TC_n})^2)}$$

Where:

- I_{MC} is the maximum current for each component on a given adapter.
- I_{TC} is the typical current for each component on a given adapter and the sum of all I_{TC} equals I_{TA}.

Note: If I_{MC} or I_{TC} is not available, estimate by using: $I_{TC} = 0.7 \times I_{MC}$

The total channel current is also determined in a statistical manner. Total Channel Current =

$$(I_{TA_1} + \cdots + I_{TA_n}) + \sqrt{((I_{MA_1} - I_{TA_1})^2 + \cdots + (I_{MA_n} - I_{TA_n})^2)}$$

Where:

- I_{MA} is a maximum statistical current for a given adapter.
- I_{TA} is the typical current for a given adapter.

Engineers designing a feature card that requires -5.0 Vdc will notice the system does not provide a -5 Vdc. This voltage will have to be generated on the card from the voltages supplied. Be sure the total current consumption for the card does not exceed the limits specified.

Storage Subsystems

Description

The optional storage subsystems for the Personal System/2 include four fixed disks, three adapters, a disk cache program, a 3.5-inch diskette drive, media compatibility and transition aids, a streaming tape drive, and an optical file. The four fixed disk models can be categorized into three levels. The first level includes a 20MB fixed disk with a 3.5-inch form factor and a low-power ST-506/412 adapter. A 5.25-inch 44MB fixed disk is utilized at the next level with a standard ST-506/412 adapter. The third level offers 70MB and 115MB files, each with a 5.25-inch form factor and an Enhanced Small Device Interface (ESDI) adapter. IBM does not support the coexistence of the ESDI and the ST506/412 adapters in the same system unit. The disk cache program allows a portion of system memory to be used as a buffer to improve the performance of application programs that read fixed disk data. A 3.5-inch, 1.44MB Diskette Drive is integrated into each Personal System/2 Model 50, 60 and 80. A second drive can be added as an option. Compatibility between 3.5-inch and 5.25-inch media is available through the use of several data transfer options such as the Data Migration Facility and the 5.25-inch External Diskette Drive. The 6157 Streaming Tape Drive is a save/restore device and the 3363 optical disk drive is a write once/read many (WORM) device.

DASD Subsystems

Fixed Disks and Adapters

The 20MB fixed disk (on Model 50) has a transfer rate of 5M bits-per-second and an average access time of 80ms. It has a 3.5-inch form factor and four data heads. The drive is formatted with 17 sectors per track and 512 bytes per sector. The sectors are numbered with a 1:1 interleave factor and a skew of 2. Defective sectors are flagged at the factory, and each physical sector location is recorded in the defect map located on the last of the 612 cylinders.

The 3.5-inch, 20MB fixed disk is attached to the Fixed Disk Drive Adapter/A Type 1 adapter through a low power version of the ST-506/412 interface. The adapter supports one drive.

The 5.25-inch, 44MB fixed disk (on Models 60-041 and 80-041(has seven data heads. It has a transfer rate of 5M bits-per-second and an average access time of 40ms. The 44MB file is formatted at the factory with the same number of sectors per track and bytes per sector as the 20MB file, but the 44MB device has 733 cylinders. Its defect map is stored on the last cylinder.

The 44MB fixed disk uses the Fixed Disk Drive Adapter/A Type 2, an adapter that is very similar to Type 1. They both support automatic head and track switching, and they both have 2Kb sector buffers. Their system interfaces support Programmed Input/Output (PIO) and Direct Memory Access (DMA) in either 8- or 16-bit modes. The Fixed Disk Drive Adapter/A Type 1 supports one 20MB drive per adapter. The Fixed Disk Drive Adapter/A Type 2 supports two drives per adapter. Type 2 is based on a standard ST-506/412 interface.

The 5.25-inch, 70MB fixed disk (on Models 60-071 and 80-071) has a transfer rate of 10M bits-per-second and an average seek time of 30ms. It uses four 5.25-inch disks and has seven data heads. The 70MB file is formatted with 583 cylinders and 36 sectors having 512 bytes per sector.

The 5.25-inch, 115MB fixed disk (on Model 80-111) has a 10M bits-per-second transfer rate and an average seek time of 28ms. It also has four disks and seven data heads. The 115MB file is formatted with 915 cylinders, 36 sectors per track, and 512 bytes of user data per sector.

Both the 70MB and 115MB fixed disks use the The ESDI Fixed Disk Drive Adapter/A adapter. This adapter supports two drives through the ESDI with a 10M bits-per-second data rate. It uses DMA to handle data transfers between the drive and the system. During burst mode DMA, a sixteen word FIFO is employed and the burst lengths are programmable.

The ESDI Fixed Disk Drive Adapter/A uses several techniques to optimize performance. Data transfers are based on a 1:1 interleave factor to improve access time for those application programs that request data from multiple sequential sectors. The adapter's 16 Kb sector buffer can store almost a full track, and transfers involve a single I/O wait state.

The ESDI Fixed Disk Drive Adapter/A uses command stacking to support multi-tasking for all attached fixed disks. A Relative Block Address (RBA) architecture is used at the BIOS interface to implement a high level, device independent interface. An on-board BIOS supports interrupt sharing and is stored in ROM on the adapter. The ESDI Fixed Disk Drive Adapter/A offers automatic defect management through the Format command. Additional media defects can be identified by requesting an extended surface analysis, and defect maps can be updated.

The ESDI Fixed Disk Drive Adapter/A enhances the reliability of the DASD subsystem because it employs a high level of VLSI integration and uses an extensive set of ROM-based diagnostics. Errors can be detected during format, read, write, and positioning operations. The adapter has the ability to retry certain operations after an error is detected. If the retry is successful, the command is executed without system intervention.

Disk Cache

Disk Cache improves the performance of application programs by using a portion of system memory as a buffer for the fixed disk. It allows existing applications to take maximum advantage of 1:1 interleaving. It decreases application execution time by using the cache buffer to store fixed disk data that is frequently accessed. A device driver sends the data to the application program from the cache buffer much quicker than if it had been read from the fixed disk.

Only one device driver is required to support the Disk Cache. When it is installed, the cache size can be selected. If low memory is used, the valid range is 16Kb to 512Kb. The range for extended memory is 16Kb to 15360Kb. The default size is 64Kb for low memory and 128Kb for extended memory. Since some application programs may perform better with different page size values, the cache page size can be specified as 2, 4, or 8 pre-fetched sectors.

3.5 Inch Diskette Drive and Media

The internal 3.5-inch diskette drive uses media with unformatted capacities of either 1MB or 2MB. These correspond to formatted capacities of 720Kb and 1.44MB, respectively.

The drive is a direct access device containing a spindle drive, head positioning mechanism, and read/write/erase logic. The spindle drive spins the diskette at a constant speed of 300 revolutions per minute. The two read/write heads are positioned over the desired track by a stepper motor. Both the 1MB and 2MB media have 80 tracks per side with a density of 135 tracks per inch.

The 1.44MB drive's dual mode capability enables the Personal System/2 family to be compatible with systems that use the 720Kb drive. Two-way interchangeability is accomplished through the 1MB mode. The 1.44MB drive and the 720Kb drive can format, read, write, or erase the 1MB diskettes. The 1MB media should be formatted by the 1.44 MB drive only when the drive is in the 1MB mode. In this mode, diskettes are formatted with 9 sectors per track and a capacity of 720Kb.

The 2MB mode is the default mode for the 1.44MB drive. The 2MB media is formatted with 18 sectors per track and a 1.44MB capacity. The 2MB media should be used only in the 1.44MB drive and only when the 1.44MB drive is in the 2MB mode.

Media Compatibility and Transition Aids

Description

There are four mechanisms available to provide transition and compatibility between the 5.25-inch and 3.5-inch media and to transfer data between the Personal Computer and Personal System/2 families. They include The Data Migration Facility for system conversion or occasional data transfer and The 5.25" External Diskette Drive for frequent data exchange. Frequent exchanges can also be accomplished by using an external 3.5-inch diskette drive connected to a Personal Computer system or through a network connection between Personal Computer and Personal System/2 units.

The Data Migration Facility consists of a cable adapter and a Send/Receive program. It uses the Personal Computer printer port to establish a unidirectional parallel data path from the Personal Computer to the Personal System/2 system. The Send program is available on a 5.25-inch diskette for use on the Personal Computer system. The Receive program is included on the Personal System/2 Reference Diskette.

The 5.25" External Diskette Drive provides for bidirectional data exchange between a Personal System/2 unit and a 5.25-inch diskette. It can be used to access programs and data stored on 5.25-inch

media without having to convert to 3.5-inch diskettes. It can also be used as an additional storage device when frequent data transfers are required.

The 5.25" External Diskette Drive consists of an external 5.25" 360Kb diskette drive, adapter card, feature card, and two cables. Software support is provided through BIOS and DOS. No additional DOS device drivers are required. The drive is compatible with all DOS commands and is not bootable. The 5.25" External Diskette Drive is a half high drive with an enclosed power supply. The adapter attaches to the connector for the 3.5-inch "B" drive. The feature card attaches to a feature slot and an internal cable connects the two cards.

External Devices

Description

The Personal System/2 external storage devices include a streaming tape and an optical file. The 6157 tape is a save/restore device. The optical file is an archiving and information distribution device that is also a reliable save/restore device.

IBM 6157 Streaming Tape Drive

The 6157 is a 55MB streaming tape drive that uses a 0.25-inch cartridge. It offers a 5M bits-per-minute burst rate and a tape speed of 90 inches per second. The 6157 supports an 8K bits per inch recording density. It is a standalone unit with its own power supply. Software support includes a menu-driven save/restore utility.

IBM 3363 Optical Disk Drive

The 3363 Optical Disk Drive is a high capacity disk storage device with removable media. It utilizes the high density characteristics of optical storage technology to offer 200MB of online capacity per drive. The removable media is a 130 mm (5.25-inch) WORM disk, housed in a protective cartridge. The cartridge can be pre-recorded with data so that large text or data bases can be distributed on a media that can be updated as the data changes. Media archive life is greater than ten years.

The Optical Disk attaches to most current IBM PCs and to all models of the Personal System/2 family via an adapter. Each adapter supports two drives. Depending on the model, from one to four adapter cards can be supported, yielding from 200MB to 1.6Gb of removable, write once, on-line storage. Data is read or written using a laser read/write head and a spiral format consisting of 512 byte sectors, 23 sectors per track, and 17,100 tracks. Average seek time across one-third of the disk surface is less than 230ms. A shorter seek time of 45ms is possible if the seek is within 51 tracks of the current head position. This is accomplished by changing the angle of the laser read/write head.

A streaming mode can be used to read up to 64Kb of sequential data with a single command. In this mode, the Optical Disk offers an effective transfer rate of 171Kbits per second.

The 3363 Optical Disk Storage Unit is supported by software which provides file access extensions to DOS, a save/restore function and a diagnostic program. The file access support allows the creation of a large number of files and files greater than 32MB in size. The commands look like standard DOS commands. The Optical Disk support software incorporates a directory structure that optimizes directory search speed and disk space utilization. Compatibility is maintained with all supported DOS commands. The user can keep back versions of the same name file. A new set of commands is used to manipulate the versions. The save/restore function allows the user to save only changed data files, while maintaining the integrity of the directory structure. Restoring files can be accomplished by file name, a particular version, all versions, or all files. This save/restore function and the multiple version support combine to give the Optical Disk an automatic audit trail capability.

Integrated Features

The Personal System/2 family of computers features a high level of system board integration.

The features integrated on the Personal System/2 are:

1. Processor Support Functions
2. Direct Memory Access (DMA)
3. System Board Input/Output (PIO) Function
4. Interrrupts
5. Numeric Co-processor
6. Serial Port
7. Parallel Port
8. Keyboard/Pointing device Controller.
9. CMOS RAM
10. System RAM
11. Diskette interface logic
12. Video

DMA Function

The Personal System/2 Direct Memory Access (DMA) controller allows high and low speed I/O devices to transfer data directly to and from memory. This allows a higher CPU throughput by freeing the processor of the I/O chores. The DMA chip supports 8 channels. The DMA chip has a programming interface compatible to the Intel 8237 Programmable DMA controller with added enhancements provided by an extended interface. There is however a fundamental difference in operation between the Personal System/2 DMA chip and the 8237 chip. The Personal System/2 DMA controller supports the new Micro Channel Architecture and uses a serial method to transfer data, while the 8237 DMA used in previous IBM personal computers uses a parallel DMA interface.

The processor can address the DMA and read or modify the controller's internal registers to define the various DMA modes, transfer addresses, transfer counts, channel masks and page registers.

The functions of the DMA controller can be grouped into two categories:

1. Program Condition
2. DMA Transfer.

During Program Condition, the DMA registers may be programmed or read using I/O instructions. Program Condition commences when the processor references the DMA with an address in the DMA I/O Address Space.

The second functional category of the controller is the actual DMA Transfer. A DMA transfer is initiated when a DMA slave wins the Arbitration bus and the DMA controller has been programmed to service the winning request in process. DMA Transfer may be in the form of a single transfer, multiple transfer (BURST), or read verification. During transfers, one byte or one word can be passed every (370 + (600)N) nanoseconds where N is the number of transfers. By asserting IOCHRDY slower I/O or memory devices can extend the I/O transfer cycle. During read verification, one byte is read from memory and no transfer to I/O occurs.

DMA Controller Transfer Modes

1. Data transfers between memory and I/O devices.

   Transfers are carried out serially, this means data is read from a device and latched in the DMA chip before it is written back to a second device. The memory address (target address) need only be specified for the DMA data transfer. The I/O address will be forced to hex 0000 during the I/O transfer. An optional 16-bit I/O address can be provided during the I/O portion of the transfer

2. Read verification operations.

   The DMA chip does a memory read operation without doing an I/O operation. The address, count will be modified and Terminal Count will be provided.

3. Memory Refresh cycles.

   The DMA controller runs a refresh cycle when requested and provide the nine bit address and control status for the operation.

General DMA Enhancements

In addition to being software compatible to the IBM Personal Computer AT, the DMA chip will support the following additional functions not found on the Personal Computer AT:

- Readable Mode registers.
- One additional DMA channel (channel 4).
- Programmable bus width (8 or 16 bits).

- Virtual DMA channels 0 and 4.
- Page register crossover.
- Extended program control operations.
- Compatible with Micro Channel Architecture

DMA Extended Mode

The Personal System/2 DMA controller features an extended mode which allows the DMA controller to be programmed via a simpler and more coherent programming interface. This mode provides additional read/write ports without the need to increase the I/O space requirements. Programming the DMA controller using the extended mode is done by using a Function Register and I/O addresses 018H and 01AH. The extended mode also allows access to registers which were inaccessible in previous IBM Personal Computers.

```
PROGRAM COMMAND                  I/O ADDRESS

Function Register .............(FR)..... 18H
Execute Function Register ... (EFR) ... 1AH
```

Figure 72. DMA Extended Register

Extended Mode Register

The DMA supports an Extended Mode Register for each channel which can be programmed and read back by the processor. This register is resident in the DMA RAM and activated whenever a DMA channel requests DMA data transfer.

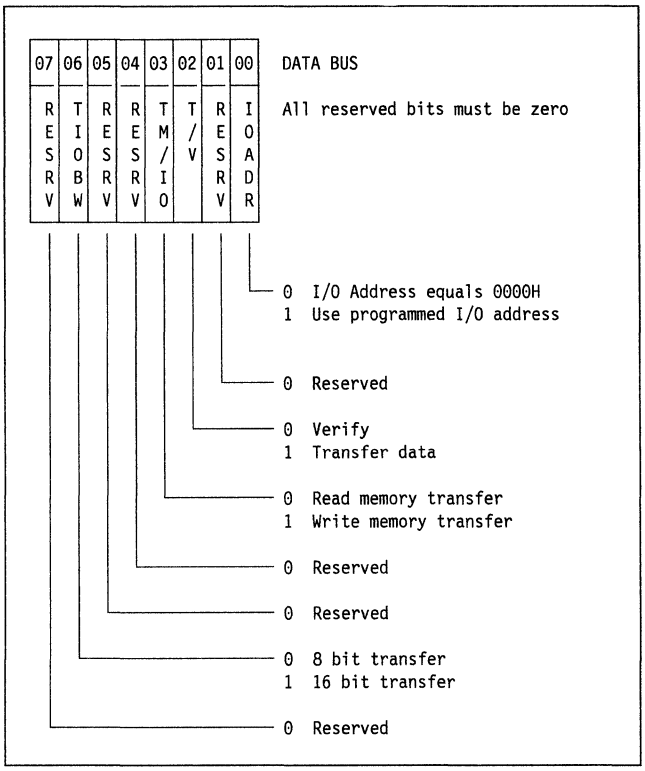


Figure 73. Extended Mode Register

Programming the DMA Controller Using Extended Mode

In order for the processor to read or write any of the DMA's internal registers, the processor will have to execute one or both of the following steps.

1. Write the Function register by executing an OUT to address 18 hex with the proper data to indicate the function and the channel number. The internal byte pointer register will always reset to 0 when the OUT to port 18H is detected. Some functions are implemented directly and do not require additional steps.

   The functions selected via the function register are shown below.

FUNCTIONS	BITS	FUNCTION HEX	DIRECT CMD	RD	WR
I/O Address Register	00-15	0	.....	X	X
Reserved		1			
Reserved		E			
Reserved		F			
Memory Address Register (Write)	00-23	2	.....	X	X
Memory Address Register (Read)	00-23	3	.....	X	
Transfer Count (Write)	00-15	4	.....	X	X
Transfer Count (Read)	00-15	5	.....	X	
Status Register Read	00-07	6	.....	X	
Mode Register	00-07	7	.....	X	X
Arbus Register	00-07	8	.....	X	X
Mask Register Set Single Bit	.....	9	 X		X
Mask Register Reset Single Bit	.....	A	 X		X
Reserved	.....	B	 X		X
Reserved	.....	C	 X		X
Master clear	.....	D	 X		X

Figure 74. DMA Extended Program Functions

Function	Channel #	DATA BUS
7 4	3 0	

Figure 75. Function register (Port 18H)

As shown in the figure above, the channel to be programmed is selected on data bits 0-3, and the function is programmed on data bits 4-7

2. Execute the function by doing an IN or OUT to address 1AH. The byte pointer will automatically increment by one and point to the next byte every time the 1AH port is used. Direct commands do not need this second step because the function will be executed upon detection of the OUT to address 18H.

Nineteen different functions on 8 different channels are implemented by the DMA by only using two I/O ports. This register can support up to 15 DMA channels plus additional functions. Port 19H and 01BH are reserved by IBM for future use.

Programming Example Using the Extended Mode

The purpose for the DMA extended operation is to make DMA programming easy and independent of the channel being programmed. The following is an example showing the programming of channel 2 using the 8237 and the extended mode.

Program step	8237 Mode OUT to ADRS Data	Extended Mode OUT to ADRS Data
Set Channel Mask Bit	(0Ah) x6h	(18h) 92h
Clear Byte Pointer	(0Ch) xxh	(18h) 22h
Write Target Address	(04h) xxh	(1Ah) xxh
Write Target Address	(04h) xxh	(1Ah) xxh
Write Page Register	(81h) xxh	(1Ah) xxh
Clear Byte Pointer	(0ch) xxh	(18h) 42h
Write Byte Count	(05h) xxh	(1Ah) xxh
Write Byte Count	(05h) xxh	(1Ah) xxh
Write Mode Register	(0Bh) xxh	(18h) 72h
		(1Ah) xxh
Reset Channel 2 Mask Bit	(0Ah) x2h	(18h) A2h

Figure 76. Channel 2 Programming Example

The following is an example showing the programming of channel 5 using the 8237 and the extended mode.

Program step	8237 Mode OUT to ADRS Data	Extended Mode OUT to ADRS Data
Set Channel Mask Bit	(D4h) x5h	(18h) 95h
Memory Address Register Write	(D8h)	(18h) 25h
Write Memory Address	(C4h) xxh	(1Ah) xxh
Write Memory Address	(C4h) xxh	(1Ah) xxh
Write Page Register	(8Bh) xxh	(1Ah) xxh
Transfer Count Register Write	(D8h)	(18h) 45h
Write Word Count	(C6h) xxh	(1Ah) xxh
Write Word Count	(C6h) xxh	(1Ah) xxh
Write Mode Register	(D6h) xxh	(18h) 75h
		(1Ah) xxh
Reset Channel 5 Mask Bit	(D4h) x1h	(18h) A5h

Figure 77. Programming Example using Extended Mode

It is clear from the above examples the extended method has consistency and can easily be adapted to any channel.

Virtual DMA Channel operation

This feature creates additional DMA channels at the arbitration levels 08H-0EH using the two 4 bit register called Arbus Registers. These registers provide the processor a way to dynamically re-assign the arbitration ID value to which the DMA controller will respond to the bus arbitration for DMA requests. This allows channels 0, 4 to service devices at any arbitration levels. The value of Fh is reserved for the processor and must never be used. In order to program the Arbus registers, the extended function 8 should be used. The upper 4 bits of the function register should have 8 and the lower eight bits the channel number (0 or 4).

Although channels 0, 4 can respond to arbitration levels other than 0, 4, the programming of these two channels still has to be done as 0, 4. In other words the arbitration level of the channel and the programming level of the channel are independent.

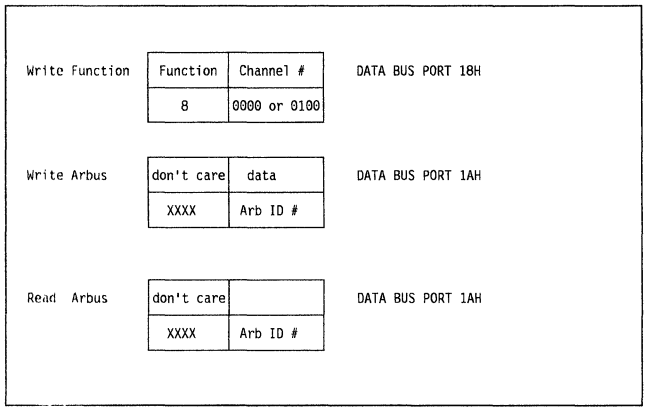


Figure 78. Arb bus Register Programming

PIO Function

The system board Input/Output functions are:

- Timer subsystem
- System Board Programmable Option Select Logic
- Port B and NMI mask Logic
- Alternate Processor Reset
- I/O selects for various system board functions.

Timer Subsystem

Three Counter/Timer elements are contained within the PIO chip. They are Channel 0, Channel 2 and Channel 3. Channel 0 and Channel 2 are same as Channel 0 and Channel 2 of the IBM PC, IBM PC XT, and the IBM Personal Computer AT. Channel 3 implements the Watchdog function and has no counterpart in previous IBM personal computers.

Channel 0 System Timer

Channel 0 is used a general purpose timer providing a constant time base for implementing a time of day clock. Programming this timer is same as in previous IBM Personal Computers.

Channel 2

Channel 2 is used to support tone generation for the audio speaker. Programming this timer is same as in previous IBM Personal Computers.

Channel 3 Watchdog Timer

The Watchdog Timer is a new feature of the Personal System/2. This feature can be used to detect error conditions, such as a hung system.

The watchdog timer detects when IRQ0 is active for more than one period of the output of timer 0 (CLK OUT 0). If IRQ0 is active when a rising edge of CLKOUT 0 occurs, the count is decremented. When the count is decremented to 0 a NMI is generated. Thus, the Watchdog timer may be used to detect when IRQ0 is not being serviced.

BIOS interfaces are provided to enable and disable the Watchdog timer. On timeout, a NMI is generated and bit 4 on port hex 92 is set. This bit is cleared using the BIOS interface to disable the Watchdog timer.

With the watchdog timer enabled, if IRQ0 remains pending for more than one period of CLKOUTO, any data written to channel 0 or Channel 3 is ignored. Channel 2 can still be written to.

Watchdog timer operation is only defined when Channel 0 is programmed in Mode 2 or Mode 3. The operation of the Watchdog timer is undefined when Channel 0 is programmed in any other mode.

Programmable Option Select Logic

The PIO chip contains logic to enable and disable system board functions. Access to the POS registers on this chip is through I/O addresses 094H and 102H.

Port B and NMI Logic

Port B is accessed by I/O Read or write operations to I/O address 0061H.

Non-maskable interrupt requests from the channel are subject to mask control from an internal latch on the PIO chip located at address 0070H(bit 7). This address is shared with the address register for the 64 byte CMOS Ram. The power on default of the NMI mask is "1" or NMI disabled.

Alternate Processor Reset

The Personal System/2 systems have an alternate means for resetting the processor This is in addition to the reset provided by the 8042 keyboard controller, as in the PC AT systems. The alternate processor reset signal is OR'd with the reset output of the 8042 keyboard controller.

This signal is pulsed high for a minimum of 100 ns when bit 0 of port 92 Hex is changed from a "0" to a "1".

System Interrupts

The Personal System/2 system board provides 16 levels of system interrupts. Any or all of the interrupts may be masked including the non-maskable interrupt. Two Intel 8259 chips provide the interrupt function.

Unlike previous IBM personal computers, the interrupt controller chips on the Personal System/2 systems are initialized to the level sensitive mode. Logic on the system boards prevents software from setting the interrupt controllers to the edge sensitive mode.

Nonmaskable Interrupt

The nonmaskable interrupt (NMI) signals to the system microprocessor that a parity error, an I/O channel check, a system bus time out, or a system Watchdog Timer time out has occurred. The non-maskable interrupt feature can be masked like in the Personal Computer AT, except the the Watch Dog Timer and bus timeout which cannot be masked. However the Watchdog Timer can be de-activated.

Interrupt Assignments

The following figure shows the interrupt assignments, interrupt levels and their functions. The interrupt levels are listed in the figure by order of priority. The highest priority is the NMI and the lowest is IRQ7.

Level	Function		
NMI	Parity, Watchdog timer Arbitration timeout Coprocessor, Channel Check		
IRQ 0	Timer		
IRQ 1	Keyboard		
IRQ 2	Cascade Interrupt Control	Level	Function
		IRQ8	Real Time Clock
		IRQ9	VRTC EGA
		IRQ10	Reserved
		IRQ11	Reserved
		IRQ12	Mouse
		IRQ13	Coprocessor Error
		IRQ14	Fixed Disk
		IRQ15	Reserved
IRQ3	Serial Alternate		
IRQ4	Serial Primary		
IRQ5	Reserved		
IRQ6	Diskette		
IRQ7	Parallel Port		

Figure 79. Personal System/2 Family Interrupt Assignment

Numeric Coprocessor

The Personal System/2 system boards provide a socket for installing a Math Coprocessor.

In the Model 50 and Model 60 systems a 80287 co-processor is used.

The Model 80 system uses a 80387 coprocessor.

The 80287 and 80387 are software compatible with the processors used on previous IBM Personal Computers. They enable the system to perform high speed arithmetic, logarithmic and trigonometric operations. The coprocessor runs at the same speed as the processor in the Personal System/2 systems (Models 50, 60, 80-041 and 80-071). Software using this option should see a significant improvement in performance over previous IBM personal computers.

Serial Port

The serial port on the system board is hardware and software compatible with previous implementations. The port can be configured to COM1 or COM2 using programmable option select. Writing a "1" to port 102H bit 3 with the port in set up mode selects COM1 and writing a "0" selects COM2.

Parallel Port

The system board parallel port allows the attachment of various devices which transfer 8 bit parallel data. In addition to being compatible with previous IBM parallel port implementations, the parallel port has an extended mode which allows bi-directional input and output.

The parallel port can be configured to three different addresses spaces previously used in IBM personal computers. These addresses are selected via the programmable option select feature. Writing to port 94H, with bit 7 set to "0", places the system board in setup mode. Writing to bits 5 and 6 at port 102H, selects the required address space.

Parallel Port Extended Mode

The extended mode option is selected via Programmable Option Select. With the system board in setup mode, extended mode is selected by writing a "0" to bit 7 at I/O Address 102. The direction of the 8-bit data port (Port A) is controlled by bit 5 in Port C. Port C has 5 bits which are compatible with previous implementations. The 6th bit corresponds to the direction control bit and is only applicable in Extended Mode.

Figure 80 shows the possible configurations for the parallel port in extended mode.

Port Direction	POS Mode Bit	Port C Direction Bit	System Reset
Write	0	0	1
Write	0	0	0
Read	0	1	0

Figure 80. Parallel Port Configurations using Extended Mode

Keyboard Controller

The 8042 was originally used in the PC AT system primarily as the keyboard interface. In the Personal System/2 systems the 8042 controller will in addition to providing the serial interface to the keyboard, will also provide a serial interface to a pointing device.

The 8042 also provides a keyboard lockout security mechanism.

Devices which provide an interface compatible with the 8042 serial interface can be used as auxiliary devices. Some examples of such a device may include:

- Mouse
- Touchpad
- Keyboard
- Trackball.

Keyboard Password Security

The 8042 provides for a password security mechanism. Three commands are used in conjunction with enabling, loading and testing the password

The commands are:

- A4 Test Password Installed
- A5 Load Security
- A6 Enable Security.

8042 Command and Status Bytes

Bit	Function
7	Reserved=0
6	IBM Compatibility mode
5	Disable Pointing Device
4	Disable Keyboard
3	Reserved=0
2	System Flag
1	Enable Pointing Device Interrupt
0	Enable Keyboard Interrupt

Figure 81. 8042 Command Byte Bit Definitions

Bit	Function
7	Parity Error
6	Generate Time Out
5	Aux Output Buffer Full
4	Inhibit Switch
3	Command/Data
2	System Flag
1	Inhibit Buffer Full
0	Output Buffer Full

Figure 82. 8042 Status Byte Bit Definitions

8042 COMMANDS (I/0 Address 064H)

The table below shows the new commands that have been added to the existing 8042 commands to support the pass word security feature and the auxiliary device

60 Write 8042 Command Byte. The next byte of data written to I/O Address 60 is placed the controller's command byte. Bit definition for bits 7, 6, 5, 3, 2, and 0 remain the same as the IBM personal Computer AT.

The following bits have been added to the command byte to support the Auxiliary device

Bit 5 - Disable Auxiliary Device - Writing a 1 to this bit disables the Auxiliary device interface by driving the "clock" line low. Data is not sent or received

Bit 1 - Enable Auxiliary device Output Buffer Full Interrupt Writing a 1 to this bit causes the 8042 to generate an interrupt when it places Auxiliary device data into its output buffer

A4 Test Password Installed - This command checks if there is currently a password installed in the 8042.

A5 Load Security - This command initiates the Password Load Procedure

A6 Enable Security - This command enables the Security Feature

A7 Disable Auxiliary device Interface - This command disables the Auxiliary device

A8 Enable Auxiliary device

A9 Interface Test - This commands the 8042 to test the Auxiliary device clock and data lines

D2 Write Keyboard Output Buffer

D3 Write Auxiliary Device Output Buffer.

CMOS RAM

The Personal System/2 system board has a Real Time Clock Chip which provides system with time of day, date and 50 bytes of user RAM. The 50 bytes of user RAM are used for storage of system configuration information.

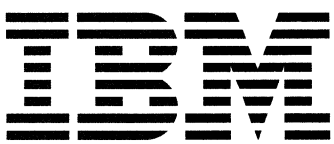
The Model 60 and Model 80 systems have an additional 2K of battery backed CMOS RAM.

IBM Personal System/2 Seminar Proceedings

Publication Number	Vol.	Topic
G360-2653	V5.1	IBM Personal System/2 Model 30 IBM Personal Computer DOS, Version 3.30
G360-2678	V5.2	IBM Personal System/2 Displays and Display Adapters
G360-2637	V5.3	IBM Personal System/2 Models 50, 60, 80 Micro Channel™ Architecture, Hardware Features and Design Considerations

G360-2637

IBM Corporation
Editor, IBM Personal System/2 Seminar Proceedings
Internal Zip 3636
Post Office Box 1328
Boca Raton, Fl 33429-1328



Design Considerations for Micro Channel™ Expansion Memory

General Post Flow

In a configured system with no errors, POST enables all adapters and tests the translation functions of cards with the supported expansion memory card design. This design is described in the *80286 Memory Expansion Option Technical Reference*. POST then counts and tests all enabled memory and compares the amount of memory it has counted to the amount of memory stored in CMOS when the system was configured by the setup utility on the Reference Diskette. These setup values are stored in the CMOS CRC areas.

Partial Flow for POST

```
For each memory card (ID = FEFE) do
     Test Translation RAM
     Set Translation RAM using CMOS parameters
end do

If CMOS valid and configuration ok then
Enable all card slots except memory

Count memory above 64K
Update CMOS locations 035H and 036H with memory
  above 640 count
Test memory 64K to 640K
Test memory above 1 Meg
If memory above 1 Meg is not equal to expected
  memory display 164 error
Test other devices
Scan for ROM in IO channel
Boot from Diskette and Disk
```

Setup Considerations

The setup program on the Reference Diskette only recognizes the memory which is on adapter cards that match the supported memory expansion adapter design. The setup program determines the amount of expansion memory and stores this value at 017H and 018H in the CRC area in CMOS. It is stored in 1K units.

If there is a miscompare between the amount of memory counted by POST and the amount stored by the setup program in the CRC area, a 164 "memory size error" is posted. If the CRC is incorrect, a 162 "configuration error" is displayed by POST. Memory cards which do not want their memory to cause a miscompare must disable it. Cards containing expansion memory of a design other than the supported one can enable and test their memory after POST to avoid these problems.

The amount of memory counted by POST is stored in the CMOS locations 035H and 036H. These locations are outside the CRC area and can be updated without a POST error. The value represents the number of 1K blocks of memory above 1 Meg. The BIOS interrupt 12H, Memory Size Determination, uses the values in CMOS locations 035H and 036H.

Adapters containing multiple functions are setup if one of the functions is not expansion memory and if a correct ADF file for the adapter is merged onto the Reference Diskette. Combination cards containing expansion memory must setup the memory and update CMOS locations 035H and 036H after POST.

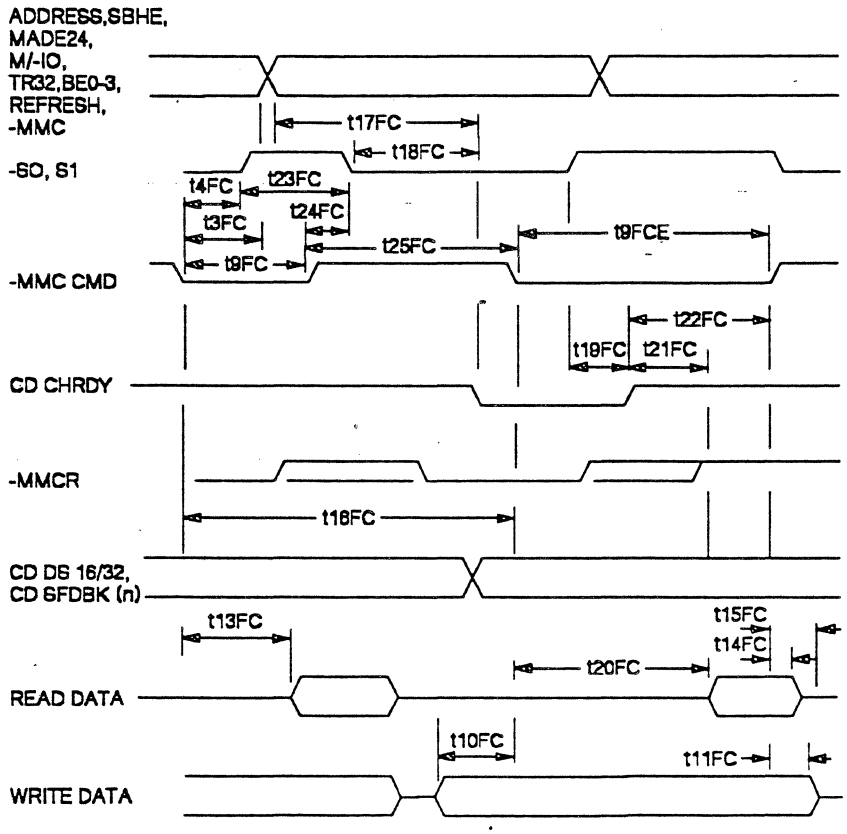
THE PUBLICATION OF THE INFORMATION CONTAINED HEREIN IS NOT INTENDED TO AND DOES NOT CONVEY ANY RIGHTS OR LICENSES, EXPRESS OR IMPLIED, UNDER ANY IBM PATENTS, COPYRIGHTS, TRADEMARKS, MASK WORKS OR ANY OTHER INTELLECTUAL PROPERTY RIGHTS.

Personal System/2 and Micro Channel are trademarks of the International Business Machines Corporation.

© Copyright International Business Machines Corporation 1987. All rights reserved.

No part of this work may reproduced or distributed in any form or by any means without prior permission in writing from the IBM Corporation.

Matched Memory Cycle Timing (One Wait)



Symbol	Description	Min/Max
t3 FC	ADDR hold from -MMC CMD active	20 / - ns
t4 FC	Status hold from -MMC CMD active	25 / - ns
t9 FC	-MMC CMD pulse width	85 / - ns
t9 FCE	-MMC CMD pulse width (Extended Cycle)	145 / - ns
t10 FC	Write data valid to -MMC CMD active	0 / - ns
t11 FC	Write data hold from -MMC CMD inactive	30 / - ns
t13 FC	Read data valid from -MMC CMD active	- / 60 ns
t14 FC	Read data hold from -MMC CMD inactive	0 / - ns
t15 FC	Read data off delay from -MMC CMD inactive	- / 40 ns
t16 FC	-MMC CMD active to next -MMC CMD active	240 / - ns
t17 FC	CD CHRDY valid from ADDR valid	- / 70 ns
t18 FC	CD CHRDY valid from Status active	- / 30 ns
t19 FC	CD CHRDY active from Status inactive	- / 9 ns
t20 FC	Read data valid from -MMC CMD active	- /125 ns
t21 FC	Read data valid from CD CHRDY active	- / 45 ns
t22 FC	-MMC CMD inactive from CD CHRDY active	65 / - ns
t23 FC	Status inactive pulse width	30 / -
t24 FC	Next status active to -MMC CMD inactive	- / 5 ns
t25 FC	-MMC CMD inactive pulse width	85 / - ns

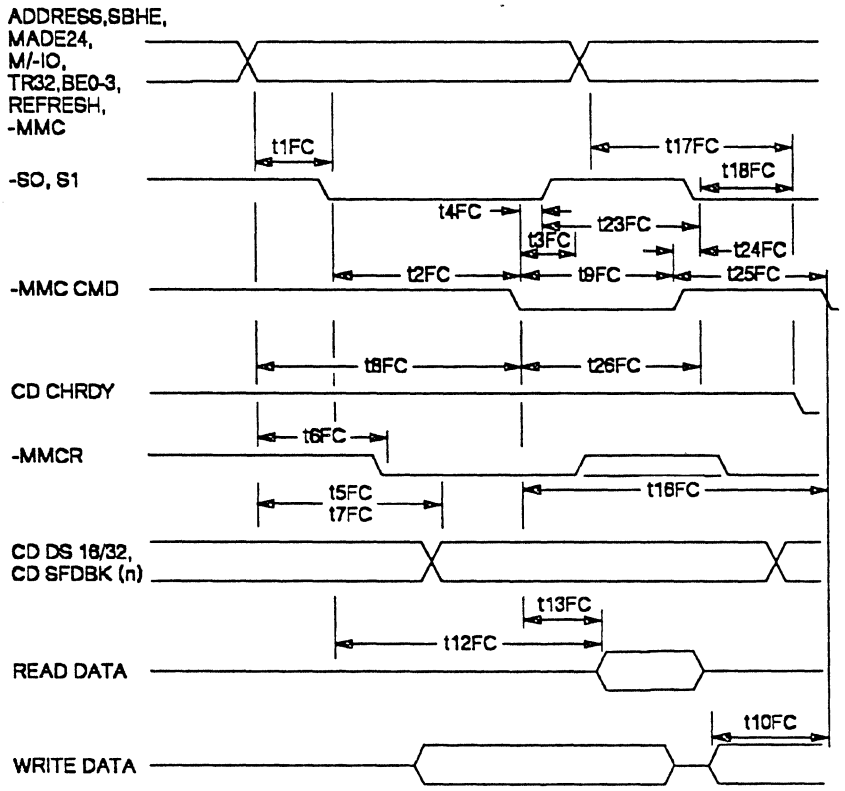
Figure 47. Matched Memory Cycle with One Wait

Matched Memory cycles support 32-bit and 16-bit channel devices only. Eight-bit channel devices cannot run Matched Memory cycles.

Note: Adapters using this protocol can only be installed in the three 32-bit channel connectors and therefore sacrifice some *portability*. These adapters may not fit or operate in other systems.

When MMC is active, Matched Memory cycle 32-bit and 16-bit devices should not use -MADE 24, A0, A1, or -SBHE in logic that generates -MMCR, -CD DS 16/32, -SEL FBK, and -CD CHRDY.

Matched Memory Cycle Timing (No Waits)



Symbol	Description	Min/Max
t1 FC	ADDR valid to Status active	10 / - ns
t2 FC	Status valid to -MMC CMD active	82 / - ns
t3 FC	ADDR hold from -MMC CMD active	20 / - ns
t4 FC	Status hold from -MMC CMD active	25 / - ns
t5 FC	CD DS 16/32 active from ADDR valid	- / 55 ns
t6 FC	-MMCR active from ADDR valid	- / 55 ns
t7 FC	-CD SFDBK active from ADDR valid	- / 60 ns
t8 FC	ADDR valid to -MMC CMD active	100 / - ns
t9 FC	-MMC CMD pulse width	85 / - ns
t10 FC	Write data valid to -MMC CMD active	0 / - ns
t11 FC	Write data hold from -MMC CMD inactive	30 / - ns
t12 FC	Read data valid from Status active	- /145 ns
t13 FC	Read data valid from -MMC CMD active	- / 60 ns
t14 FC	Read data hold from -MMC CMD inactive	0 / - ns
t15 FC	Read data off delay from -MMC CMD inactive	- / 40 ns
t16 FC	-MMC CMD active to next -MMC CMD active	180 / - ns
t17 FC	CD CHRDY valid from ADDR valid	- / 70 ns
t18 FC	CD CHRDY valid from Status active	- / 30 ns
t23 FC	Status inactive pulse width	30 / - ns
t24 FC	Next status active to -MMC CMD inactive	- / 5 ns
t25 FC	-MMC CMD inactive pulse width	85 / - ns
t26 FC	-MMC CMD active to Status active	90 / - ns

Figure 46. Matched Memory Cycle (No Waits)

Matched Memory cycles support 32-bit and 16-bit channel devices only. Eight-bit channel devices cannot run Matched Memory cycles.

Note: Adapters using this protocol can only be installed in the three 32-bit channel connectors and therefore sacrifice some *portability*. These adapters may not fit or operate in other systems.

When MMC is active, Matched Memory cycle 32-bit and 16-bit devices should not use -MADE 24, A0, A1, or -SBHE in logic that generates -MMCR, -CD DS 16/32, -SEL FBK, and -CD CHRDY.

Micro Channel Connector (32-Bit Extension)

This connector extends the 16-bit Micro Channel connector to accommodate 32-bit addressing and 32-bit data transfers.

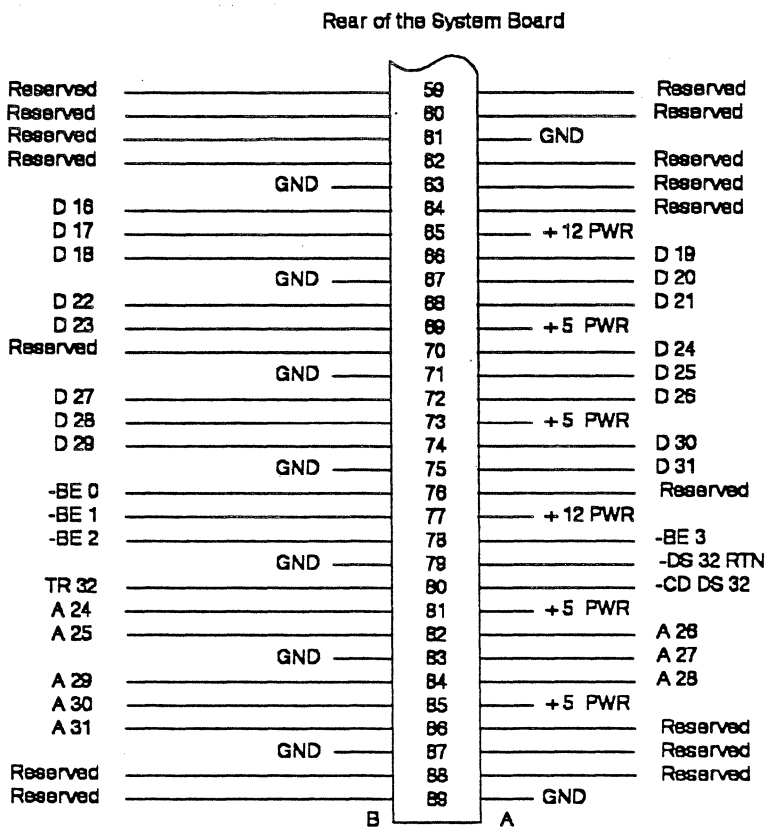


Figure 26. Micro Channel Connector 32-Bit Extension Voltage and Signal Assignments

Micro Channel Connector (Matched Memory Extension)

This connector extends the 32-bit Micro Channel connector to accommodate Matched Memory cycles.

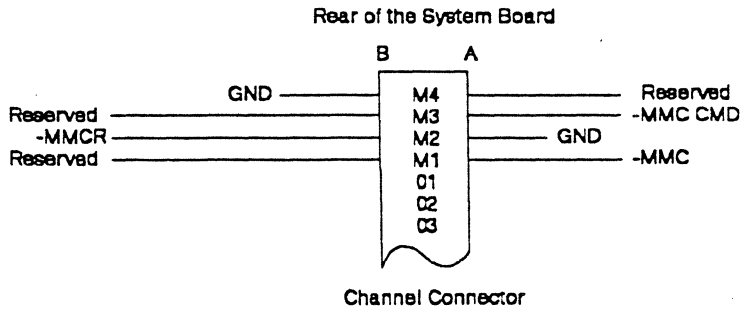


Figure 27. Micro Channel Connector Matched Memory Extension Voltage and Signal Assignments

connector (one independent signal line per connector). -CD DS 16 is driven with a totem-pole driver.

-DS 16 RTN: -Data Size 16 Return: This output signal is a negative OR of the -CD DS 16 signals from each channel connector. If any device drives its -CD DS 16 active then this output is active. This signal is provided to allow channel resident bus masters (a device that arbitrates for, and controls the channel) to monitor the data size information. -DS 16 RTN must be driven with a bus driver.

-SBHE: -System Byte High Enable: This line indicates and enables transfer of data on the high byte of the data bus (D8 - D15), and is used with A0 to distinguish between high byte (D8 - D15) and low byte (D0 - D7) transfers. All 16-bit slaves decode this line but 8-bit slaves do not. -SBHE is driven with a tri-state driver.

MADE 24: Memory Address Enable 24: This line indicates when an extended address is used on the bus. If a memory cycle is in progress, MADE 24 inactive indicates an extended address space greater than 16M is being presented; MADE 24 active indicates an unextended address space less than or equal to 16M is being presented. This line is driven by the system microprocessor or bus master and decoded by all memory slaves, regardless of their address space size.

M/-IO: Memory/-Input Output: This signal distinguishes a memory cycle from an I/O cycle. When this signal is high, it indicates a memory cycle is in progress. When M/-IO is low, it indicates that an I/O cycle is in progress. M/-IO is driven with a tri-state driver.

-S0, -S1: -Status Bits 0 and 1: These lines indicate the start of a channel cycle and also define the type of channel cycle. When used with M/-IO, memory read/write operations are distinguished from I/O read/write operations. When M/-IO, -S0, and -S1 are all high, they do not signify a status cycle. These signals are latched by the slave, as required, using the leading edge of -CMD or the trailing edge of -ADL. -S0 and -S1 are driven with a tri-state driver.

Decoded commands are generated with latched status lines (-S0 and -S1), M/-IO, and -CMD.

Slaves must support a full decode of -S0 and -S1. The following figure shows the proper states of M/-IO, -S0, and -S1 in decoding I/O and memory read/write commands.

M/-IO	-S0	-S1	Function
0	0	0	Reserved A
0	0	1	IO Write
0	1	0	IO Read
0	1	1	Reserved B
1	0	0	Reserved C
1	0	1	Memory Write
1	1	0	Memory Read
1	1	1	Reserved D

Figure 22. I/O and Memory Transfer Controls

A decoded I/O Write command instructs an I/O slave to store the data on the data bus. The data must be valid on the bus from the leading edge of -CMD and must be held on the bus until after -CMD goes inactive. Addresses on the bus must be valid prior to -S0 going active.

A decoded I/O Read command instructs an I/O slave to drive its data onto the data bus. The data must be placed on the bus following the leading edge of -CMD, must be valid before the trailing edge of -CMD and must be held on the bus until -CMD goes inactive. Addresses on the bus must be valid prior to -S1 going active.

A decoded memory Write command instructs the memory to read the data on the data bus. The data must be valid on the bus from the leading edge of -CMD and must be held on the bus until after -CMD goes inactive. Addresses on the bus must be valid prior to -S0 going active.

A decoded memory Read command instructs the memory to drive its data onto the data bus. The data must be placed on the bus following the leading edge of -CMD. The data must be valid before the trailing edge of -CMD, and must be held on the bus until -CMD goes inactive. Addresses on the bus must be valid prior to -S1 going active.

-CMD: -Command: This signal is used to define when data is valid on the data bus. The trailing edge of this signal indicates the end of the bus cycle. This signal provides to the slave the indication of the duration during which the data is valid on the bus. During write operations, the data is valid on the bus throughout the period that -CMD is active. During read operations, the data is valid on the bus following the leading edge but before the trailing edge of -CMD and must be held on the bus until after the -CMD goes inactive. This signal can be used by the slaves to latch the address on the bus. Latched status lines gated by -CMD provide the timing control of valid data. To ensure proper operation slaves must use transparent latches to latch address and status